BAB 1 PENDAHULUAN

1.1 Latar Belakang

Komputerisasi adalah suatu kegiatan yang sangat dibutuhkan pada kehidupan manusia. Dengan adanya kegiatan tersebut maka diperlukan suatu keamanan untuk aset yang dimiliki terutama data-data penting demi menjaga kerahasiaan data itu sendiri (Pabokory, Astuti dan Kridalaksana, 2015). Salah satu kegiatan komputerisasi adalah pengiriman *file* melalui internet yang dapat berisiko hilangnya privasi pengguna karena terdapat pihak ketiga atau yang biasa disebut sebagai *Man In The Middle* untuk mencuri data pengguna di mana pengguna memiliki berbagai data yang bersifat rahasia sehingga tidak semua orang boleh mengetahuinya. Banyak langkah yang dapat ditempuh untuk melindungi data pengguna seperti memberi *password* atau menyembunyikan *file* tersebut (Shoim, Irfan dan Pranoto, 2017). Seiring dengan berkembangnya zaman, langkah untuk melindungi *file* juga semakin meningkat yaitu dengan membuat suatu sistem untuk mengamankan *file* yang bersifat rahasia dari serangan pihak yang tidak berwenang dengan menggunakan teknik kriptografi.

Kriptografi adalah suatu ilmu yang berfungsi menjaga informasi untuk tetap aman saat dikirimkan dari satu tempat ke tempat lain yang telah ada sejak dulu. Kriptografi telah ada pada zaman Julius Caesar untuk mengamankan informasi dengan melakukan pergeseran tiap karakter di mana teknik tersebut aman pada zaman itu. Pada zaman sekarang, kemampuan komputer semakin canggih untuk mampu memecahkan teknik tersebut. Para ahli kriptografi telah membuat berbagai macam algoritme untuk mengamankan informasi, namun para *hacker* akan berusaha memecahkannya sehingga tidak sedikit pula algoritme yang telah dibuat tidak dapat mengamankan informasi sebagaimana mestinya. Oleh karena itu, hingga saat ini para ahli kriptografi masih berlomba-lomba untuk membuat algoritme kriptografi yang lebih aman (Primartha, 2013).

Sistem pengamanan data *file* yang telah dibangun terdapat pada penelitian sebelumnya oleh Rifkie Primartha pada tahun 2013 dengan menggunakan algoritme *Advanced Encryption Standard* (AES) yang diperkenalkan oleh NIST pada tahun 2001. Pada penelitian sebelumnya, peneliti menciptakan suatu perangkat lunak dengan memanfaatkan algoritme AES untuk enkripsi dan dekripsi teks dan *file*. Kelemahan pada penelitian sebelumnya terdapat pada penggunaan algoritme AES yang telah ditemukan ketidaksempurnaan dalam proses komputasinya melalui serangan dalam jangka panjang pada semua versi algoritme AES yang dilakukan oleh para riset Microsoft yaitu Andrey Bogdanov, Dmitry Khovratovich dan Christian Rechberger. Hal tersebut menjadi kelemahan terbesar AES dengan berhasilnya pemecahan proses komputasi dalam algoritme AES maka pertahanan dapat ditembus di seluruh sistem AES (Asriyanik, 2017).

Berdasarkan permasalahan yang ada pada AES, maka dibutuhkan solusi keamanan baru dalam hal mengamankan data. Banyak jenis algoritme kriptografi yang telah dibuat oleh para ahli kriptografi untuk mengamankan data. Contoh

algoritme *block cipher lightweight* terbaru adalah SPECK. Algoritme SPECK adalah salah satu algoritme simetris *block cipher* di mana hanya menggunakan satu kunci dalam pengaplikasiannya yang diciptakan oleh NSA dan dipublikasikan pada tahun 2013. *Block cipher* sendiri adalah metode kriptografi dengan memecah *plaintext* menjadi blok-blok sesuai dengan ukuran tertentu, kemudian masing-masing blok dienkripsi untuk menghasilkan *ciphertext*. SPECK memiliki ukuran blok sebesar 32, 48, 64, 96 dan 128 bits dan *key size* sebesar 64, 72, 96, 128, 144, 192 dan 256 bits (Ray beaulieu et al., 2013). Algoritme SPECK dapat memenuhi salah satu aspek keamanan informasi yaitu *confidentiality* yang memiliki arti jaminan terhadap kerahasiaan suatu data atau informasi yang hanya dapat diakses oleh pengguna yang berwenang dengan melakukan proses enkripsi dan dekripsi. Kelebihan dari SPECK adalah proses enkripsi yang lebih cepat dibanding saat menggunakan algoritme asimetris. SPECK dapat memenuhi syarat *block cipher* yang aman apabila memiliki sifat pemetaan acak di mana pemetaan acak sendiri adalah suatu fungsi di mana *output* terlihat acak dan menjadi sulit ditebak bagaimana *input* nya sehingga tidak diketahui hubungan antara *plaintext* dengan *ciphertext* dan kunci yang digunakan menjadi tidak diketahui juga (Firmanesa dan Wildan, 2016).

Berdasarkan uraian yang telah dipaparkan, maka dilakukan penelitian untuk menganalisis kinerja algoritme SPECK dalam proses enkripsi dan dekripsi *file* teks karena SPECK sebagai algoritme *block cipher* terbaru yang termasuk masih jarang diimplementasikan. Analisis dilakukan dengan menguji kinerja ketiga varian SPECK yaitu SPECK128/128, SPECK128/192, SPECK128/256. Pemilihan variasi SPECK yang dianalisis mengacu pada penelitian sebelumnya yang menggunakan algoritme AES di mana AES memiliki 3 variasi berdasarkan ukuran kuncinya yaitu AES-128, AES-192 dan AES-256. Penulisan notasi dalam SPECK128/128 berarti SPECK memiliki ukuran blok 128 bits dan ukuran kunci 128 bits (Ray beaulieu et al., 2013).

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang yang telah dijelaskan, maka rumusan masalah yang diangkat untuk menjadi acuan dalam penelitian adalah:

1. Bagaimana mengimplementasikan algoritme SPECK untuk memenuhi salah satu aspek keamanan informasi yaitu *confidentiality* pada *file* teks?
2. Bagaimana hasil analisis kinerja algoritme SPECK pada keamanan *file* teks?

1.3 Tujuan

1. Untuk mengimplementasikan algoritme SPECK dalam memenuhi salah satu aspek keamanan informasi yaitu *confidentiality* pada *file* teks.
2. Untuk menganalisis kinerja algoritme SPECK pada keamanan *file* teks.

1.4 Manfaat

1. Algoritme SPECK merupakan solusi keamanan data yang baru dalam sisi *software*.
2. Memberikan pengetahuan mengenai kinerja algoritme SPECK saat pengguna ingin melakukan enkripsi dan dekripsi data untuk mengamankan data pengguna.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Aspek keamanan yang diterapkan dalam penelitian hanya *confidentiality*.
2. Peneliti memasukkan *key* secara manual pada tahap pengujian.
3. Sistem melakukan keamanan data pada *file* teks berekstensi *.pdf.
4. Pengujian dilakukan sampai pada ukuran *file* 300 KB.

1.6 Sistematika Pembahasan

Sistematika pembahasan yang disusun untuk mencapai tujuan adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang diperlukan suatu keamanan pada *file* teks dan alasan terpilihnya algoritme SPECK yang kemudian menciptakan rumusan masalah untuk mengimplementasikan dan menganalisis kinerja algoritme SPECK. Batasan masalah dan sistematika pembahasan pada penelitian juga dibahas.

BAB II LANDASAN KEPUSTAKAAN

Bab ini melakukan pengkajian pustaka terhadap penelitian-penelitian sebelumnya dan terdapat teori-teori umum mengenai algoritme SPECK. Teori-teori berasal dari jurnal dan berbagai referensi lain yang berkaitan dengan topik masalah.

BAB III METODOLOGI

Bab ini menggambarkan struktur penelitian dimulai dari tahap identifikasi masalah, studi literatur dengan mencari berbagai referensi sesuai dengan topik penelitian, tahap perancangan, implementasi, pengujian, analisis hasil pengujian dan penarikan kesimpulan dan saran.

BAB IV PERANCANGAN

Bab ini menjelaskan tahap perancangan di mana terdapat tiga tahap yaitu perancangan implementasi algoritme, perancangan aplikasi sistem dan perancangan pengujian. Perancangan yang telah dibuat akan digunakan untuk diterapkan dalam proses implementasi dan pengujian.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB V IMPLEMENTASI

Bab ini menjelaskan mengenai implementasi algoritme SPECK dengan langkah-langkah pengerjaan dan akan ditampilkan potongan *source code* sebagai penggambaran dari implementasi. Implementasi dilakukan berdasarkan pada perancangan yang telah dibuat.

BAB VI PENGUJIAN

Bab ini menjelaskan pengujian yang dilakukan untuk mendapatkan data-data yang akan dianalisis sehingga dapat menilai hasil kinerja algoritme SPECK. Tahap pengujian mengikuti perancangan yang telah dibuat.

BAB VII ANALISIS HASIL PENGUJIAN

Bab ini menjelaskan hasil pengujian dan analisis kinerja algoritme SPECK. Hasil dari pengujian akan dianalisis untuk mengetahui kinerja algoritme SPECK dalam mengamankan data.

BAB VIII PENUTUP

Bab ini memaparkan kesimpulan yang didapatkan dari hasil analisis terhadap kinerja algoritme SPECK dengan adanya jawaban dari setiap rumusan masalah dan pemberian saran untuk pengembangan penelitian lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan kajian pustaka dan teori-teori dalam penyusunan laporan penelitian. Kajian pustaka berisi tentang penelitian-penelitian sebelumnya yang berhubungan dengan topik penelitian. Teori-teori terdiri dari hal-hal yang berkaitan dengan pengetahuan mengenai topik penelitian yaitu algoritme SPECK.

2.1 Kajian Pustaka

Kajian pustaka utama yang melandasi pengambilan topik skripsi adalah jurnal yang berjudul "Penerapan Enkripsi dan Dekripsi *File* Menggunakan Algoritma *Advanced Encryption Standard* (AES)" oleh Rifkie Primartha pada tahun 2013 menciptakan aplikasi keamanan teks dan *file* dengan menerapkan algoritme AES menggunakan bahasa pemrograman Java untuk melakukan proses enkripsi dan dekripsi. Proses enkripsi dan dekripsi antara teks dan *file* pada umumnya memiliki mekanisme yang sama dan menghasilkan waktu proses yang relatif sama.

Pada penelitian sebelumnya yang menggunakan algoritme AES untuk mengamankan data, peneliti kemudian mengembangkan aplikasi keamanan data dengan menggunakan algoritme SPECK yang merupakan algoritme *block cipher* terbaru di mana kajian pustaka yang digunakan dalam penelitian ini adalah jurnal yang berjudul "*The SIMON and SPECK Families of Lightweight Block Cipher*" oleh Beaulieu, Shors, Smith, Treatman-Clark, Weeks dan Wingers pada tahun 2013 menciptakan dua algoritme *block cipher* yaitu SIMON dan SPECK dengan tujuan untuk memenuhi kebutuhan atas *lightweight block cipher* yang aman, fleksibel dan mudah untuk dianalisis. Kedua algoritme tersebut diciptakan untuk memenuhi kekurangan yang ada pada *lightweight block cipher* sebelumnya yang sebagian besar dirancang untuk bekerja dengan baik pada satu *platform* saja dan tidak dapat memberikan kinerja yang baik pada beberapa perangkat yang lain. Algoritme SIMON dirancang untuk memiliki kinerja yang optimal pada perangkat keras dan algoritme SPECK dirancang untuk memiliki kinerja yang optimal pada perangkat lunak.

2.2 Kriptografi

Kriptografi merupakan suatu ilmu untuk mempelajari cara mengamankan informasi yang bersifat rahasia dari pihak yang tidak berwenang ketika informasi akan dikirimkan dari satu tempat ke tempat lain (Nurjanah, 2016). Kriptografi berhubungan dengan aspek keamanan informasi yaitu (Aulia dan Pratomo, 2016):

a. Kerahasiaan data (*confidentiality*): memastikan kerahasiaan data yang dikirim pengguna sehingga tidak dapat dilihat oleh pihak yang tidak berwenang.
b. Integritas data (*integrity*): memastikan bahwa data yang dikirim masih dalam keadaan utuh dan tidak mengalami perubahan apapun.

c. Ketersediaan data (*availability*): memastikan ketersediaan data dalam proses pengiriman.

Konsep kriptografi dalam mengamankan kerahasiaan informasi adalah dengan mengubah sebuah pesan yang dapat dibaca yaitu *plaintext* menjadi tidak dapat dibaca yaitu *ciphertext*. Proses perubahan dari *plaintext* menjadi *ciphertext* disebut proses enkripsi dan proses perubahan dari *ciphertext* menjadi *plaintext* disebut proses dekripsi (Primartha, 2013). Pada umumnya, kriptografi dibagi menjadi dua berdasarkan kuncinya yaitu algoritme asimetris dan algoritme simetris.

2.2.1 Algoritme Asimetris

Algoritme asimetris adalah algoritme yang memiliki dua kunci dalam pengaplikasiannya yang terdiri dari kunci publik dan kunci pribadi. Kunci publik digunakan untuk melakukan proses enkripsi dan kunci pribadi digunakan untuk melakukan proses dekripsi. Dalam konsep kunci asimetris, kunci publik akan didistribusikan karena tidak memerlukan kerahasiaan sedangkan kunci pribadi akan disimpan atau tidak didistribusikan. Setiap pengguna dapat melakukan proses enkripsi dari kunci publik tetapi untuk melakukan proses dekripsi hanya bisa dilakukan oleh pengguna yang mempunyai kunci pribadi. Contoh dari algoritme asimetris adalah Elgamal, RSA, ECC, dan lain-lain (Nurjanah, 2016).

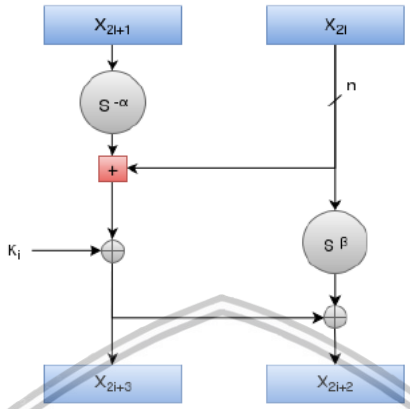
2.2.2 Algoritme Simetris

Algoritme simetris adalah algoritme yang memiliki kunci *private* yang sama dengan kunci *public* nya dalam memproses enkripsi dan dekripsi data (Nurjanah, 2016). Kunci simetris dapat dibagi menjadi dua jenis yaitu *stream cipher* yang menghasilkan *plaintext*/*ciphertext* dalam bentuk bit tunggal di mana rangkaian bit dienkripsi/didekripsi bit per bit dan *block cipher* yang menghasilkan *plaintext*/*ciphertext* dalam bentuk blok bit di mana rangkaian bit dibagi menjadi blok-blok bit dengan ukuran yang sudah ditentukan (Pratama, Nasution dan Purboyo, 2015). Contoh dari algoritme simetris adalah DES (*Data Encryption Standard*), blowfish, twofish, IDEA, 3DES, AES (*Advanced Encryption Standard*) dan lain-lain.

2.2.2.1 Algoritme SPECK

Sejak DES menjadi standar kriptografi pertama, algoritme *block cipher* telah menjadi popular dalam dunia kriptologi. Pada tahun 2013, NSA memperkenalkan algoritme *block cipher* SIMON dan SPECK. Kedua algoritme tersebut merupakan bagian dari *lightweight block ciphers.* Algoritme ini diciptakan untuk mengoptimalkan kinerja software dan hardware dalam konteks keterbatasan lingkungan *computing*. Algoritme SPECK menggunakan konstruksi sandi sederhana yang dikenal sebagai ARX (*add, rotate,* XOR). Dari berbagai variannya, SPECK mempunyai beragam ukuran data blok dan kunci. Secara umum, varian dari *cipher* disebut sebagai SPECK2n / mn dan dapat di instansiasikan dengan nilai n = 16/24/32/48/64 dan m = 2/3/4 (Togan, Lupascu dan Plesca, 2015). Di mana variabel n adalah ukuran *word* dan variabel m adalah *key word*.

Proses enkripsi pada algoritme SPECK menggunakan operasi *bitwise* XOR, modulo *addition* 2^n (+), *circular shift* ke kanan dan ke kiri (S^{-a} dan S^{-b}). Gambar SPECK *round function* ditunjukkan pada Gambar 2.1.



Gambar 2.1 SPECK *round function*

Sumber: Ray beaulieu et al. (2013)

Variabel k adalah kunci yang diproses. Untuk $k \in GF(2)^n$, pada proses enkripsi memiliki persamaan yang ditunjukkan pada Persamaan 2.1.

$$R_k(x,y) = ((S^{-\alpha} x + y) \oplus k, S^{\beta} y \oplus (S^{-\alpha} x + y) \oplus k) \quad (2.1)$$

Jika jumlah rotasi $\alpha = 7$ dan $\beta = 2$ maka $n = 16$ (*block size* = 32) dan jumlah rotasi $\alpha = 8$ dan $\beta = 3$ untuk varian ukuran *word* lainnya. Proses *inverse* pada *round function* digunakan untuk proses dekripsi dengan menggunakan *modular subtraction* (-). Proses dekripsi memiliki persamaan yang ditunjukkan pada Persamaan 2.2.

$$R^{-1}_k(x,y) = (S^{\alpha}((x \oplus k) - S^{-\beta}(x \oplus y)), S^{-\beta}(x \oplus y)) \quad (2.2)$$

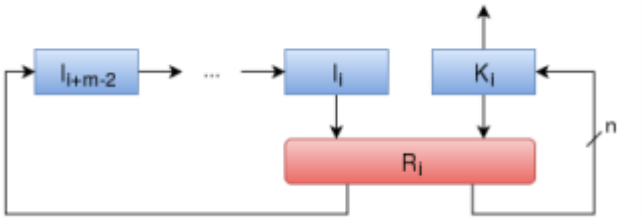
Pada SPECK *key schedule* menggunakan fungsi *round* untuk menciptakan *key* di setiap *round* nya sehingga tidak memerlukan implementasi fungsi baru. *Key schedule* menciptakan dua variabel untuk prosesnya yaitu ki dan li. Proses untuk menghasilkan nilai li ditunjukkan pada Persamaan 2.3.

$$l_{i+m-1} = (k_i + S^{-\alpha} l_i) \oplus i \quad (2.3)$$

Dan proses untuk menghasilkan nilai ki ditunjukkan pada Persamaan 2.4.

$$k_{i+1} = S^{\beta} k_i \oplus l_{i+m-1} \quad (2.4)$$

Rumus 2.3 dan 2.4 akan diproses dengan perulangan jumlah ronde yang digunakan berdasarkan jenis SPECK. Proses *key schedule* pada algoritme SPECK ditunjukkan pada Gambar 2.2.



Gambar 2.2 SPECK *key schedule*

Sumber: Ray beaulieu et al. (2013)

Algoritme SPECK memiliki varian ukuran blok dan panjang kunci yang ditunjukkan pada Tabel 2.1.

Tabel 2.1 Varian Algoritme SPECK

Block size $2n$	Key size mn	*Word* size n	Key *words* m	Rot α	Rot β	*Rounds* r
32	64	16	4	7	2	22
48	72	24	3	8	3	22
	96		4			23
64	96	32	3	8	3	26
	128		4			27
96	96	48	2	8	3	28
	144		3			29
128	128	64	2	8	3	32
	192		3			33
	256		4			34

Sumber: Firmanesa dan Wildan (2016)

2.3 Kolmogorov-Smirnov

Uji normalitas adalah salah satu uji asumsi klasik sebagai pesyaratan analisis statistik (Santoso, 2014). Tujuan dilakukan uji normalitas adalah untuk menguji nilai residual berdistribusi normal atau tidak dalam model regresi (Ghozali, 2012). Terdapat dua cara untuk mengetahui nilai residual berdistribusi normal atau tidak yaitu dengan analisis grafik dan analisis statistik. Pada analisis grafik dilakukan dengan melihat grafik histogram dengan membandingkan data yang diuji dengan distribusi yang mendekati normal. Cara kedua adalah analisis statistik dengan melakukan pengujian Kolmogorov-Smirnov.

Uji Kolmogorov-Smirnov yang paling umum digunakan karena tidak menimbulkan perbedaan pandangan terhadap hasil pengujian (Ghozali, 2012). Dalam pengujian Kolmogorov-Smirnov, apalabila nilai signifikansi < 0,05 maka data yang diuji memiliki nilai residual yang dianggap berdistribusi tidak normal. Pengambilan keputusan Kolmogorov-Smirnov sebagai berikut:

H0: nilai signifikansi > 0,05 maka data pengujian berdistribusi normal
H1: nilai signifikansi < 0,05 maka data pengujian tidak berdistribusi normal

Dalam pengujian Kolmogorov-Smirnov, untuk data sampel berukuran kecil memiliki sedikit kemungkinan menolak H0 sebaliknya untuk sampel berukuran besar (Ghasemi dan Zahediasl, 2012). Jika data yang diuji berdisribusi normal maka uji berikutnya menggunakan pendekatan parametik sedangkan jika data yang diuji tidak berdistribusi normal maka uji berikutnya menggunakan pendekatan non-parametik (Herawati, 2016).

2.4 Kruskal-Wallis

Kruskal-Wallis adalah salah satu pengujian non-parametik dalam sebuah kelompok prosedur yang ditujukan pada sampel independen. Prosedur digunakan saat membandingkan antara dua variabel yang diukur dari sampel yang tidak sama di mana kelompok yang dibandingkan lebih dari dua (Junaidi, 2015).

Pada pendekatan parametik, analisis yang dilakukan untuk membandingkan dua kelompok menggunakan analisis Anova dan sebaliknya pada pendekatan non-parametik menggunakan analisis Kruskal-Wallis. Tabel perbandingan Kruskal-Wallis dengan Anova ditunjukkan pada Tabel 2.2.

Tabel 2.2 Perbandingan Kruskal-Wallis dengan Anova

KRUSKAL-WALLIS	ANOVA
1. Data berdistribusi bebas	1. Data harus berdistribusi normal
2. Data homogen/heterogen	2. Data homogen
3. Memiliki variansi yang sama	3. Memiliki variansi yang sama
4. Sampel yang akan diuji harus independent	4. Sampel yang akan diuji harus independent

Sumber: Febriansyah (2013)

Pada penentuan kriteria pengujian, jika sampel yang diambil dengan $n_i \leq 5$ dan kelompok (k) = 3, maka tabel pembanding yang digunakan adalah tabel Kruskal-Wallis. Jika sampel yang diambil dengan $n_i > 5$, maka tabel pembanding yang digunakan adalah tabel Chi-Kuadrat (Febriansyah, 2013).

Rumus statistik untuk Kruskal-Wallis ditunjukkan pada Persamaan 2.5.

$$\boldsymbol{H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1)} \qquad (2.5)$$

Di mana :

N = jumlah sampel keseluruhan

Ri = jumlah peringkat pada kelompok i yang di*rank*ing tiap-tiap pengukuran dari seluruh datanh

K = perlakuan pada sampel

n_i = banyak sampel pengukuran pada perlakuan sampel ke i

Peringkat akan dirata-ratakan apabila menemukan sampel bernilai sama, kemudian menambahkan rumus yang ditunjukkan pada Persamaan 2.6.

$$H = \frac{\frac{12}{N(N+1)}\sum_{i=1}^{k}\frac{R_i^2}{n_i} - 3(N+1)}{1 - \frac{\sum T}{N^3 - N}} \quad (2.6)$$

Di mana:

T = $t^2 - 1$ (t adalah sampel yang berangka sama dalam nilai skor yang berangka sama)

Pengambilan keputusan yang ada pada pengujian Kruskal-Wallis untuk mengetahui adanya perbedaan waktu antara data kelompok yang diujikan dengan hipotesis sebagai berikut:

H0: nilai signifikansi > 0,05 maka tidak terdapat perbedaan antara data kelompok
H1: nilai signifikansi < 0,05 maka terdapat perbedaan antara data kelompok

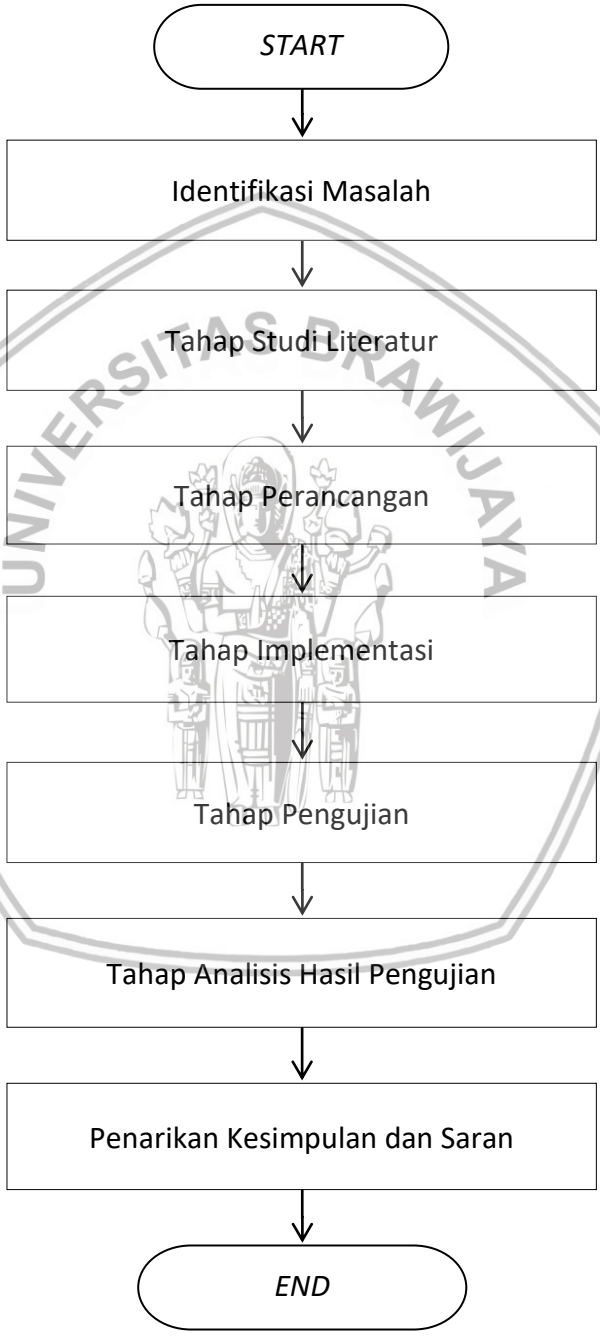
Apabila ditemukan hasil nilai signifikansi < 0,05 maka akan dilakukan pengujian *Post Hoc test* untuk mengetahui data kelompok mana yang menyebabkan perbedaan signifikan dengan melihat nilai *Chi-square* tertinggi (Salemba, 2014).

2.5 *Avalanche Effect*

Di dalam kriptografi, *avalanche effect* merupakan kondisi yang dibutuhkan algoritme kriptografi, umumnya terdapat pada *block cipher* dan *hash function*. Di dalam *block cipher* berkualitas tinggi, *avalanche effect* dapat dikatakan baik apabila perubahan sedikit pada *input key* atau *plaintext* dapat memberi perubahan yang signifikan pada *ouput*. Jika *avalanche effect* tidak menunjukkan perubahan yang signifikan, maka kriptanalis dapat memprediksi bentuk *input* hanya dengan melihat *output* saja. Hal ini juga dapat mematahkan algoritme secara sebagian hingga keseluruhan (Vadaviya dan Tandel, 2015).

BAB 3 METODOLOGI

Bab ini menjelaskan mengenai metodologi yang digunakan dalam penelitian untuk mengimplementasikan algoritme simetris yaitu algoritme SPECK. Tahapan-tahapan yang dilakukan dalam penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram alir metodologi penelitian

3.1 Identifikasi Masalah

Tahap ini melakukan identifikasi masalah berdasarkan jurnal penelitian sebelumnya dan menemukan permasalahan yang dapat digunakan sebagai bahan penelitian. Algoritme SPECK yang merupakan algoritme *block cipher* terbaru akan melakukan enkripsi dan dekripsi *file* pada penelitian sebelumnya yang menggunakan algoritme AES dalam pengaplikasiannya.

3.2 Tahap Studi Literatur

Tahap ini dilakukan pencarian referensi yang berkaitan dengan topik penelitian untuk menunjang penulisan laporan penelitian. Referensi secara tertulis dan pemahaman terhadap pengetahuan topik penelitian didapatkan dari jurnal dan berbagai sumber yang berkaitan dengan topik penelitian. Hal-hal yang terkait adalah kriptografi, algoritme simetris dan algoritme SPECK.

3.3 Tahap Perancangan

Tahap ini dilakukan proses perancangan yang dibagi menjadi tiga bagian, yaitu perancangan implementasi algoritme yang berisi manualisasi dari algoritme SPECK, perancangan sistem untuk menjelaskan bagaimana cara kerja sistem keseluruhan dan perancangan pengujian untuk menjelaskan rangkaian pengujian yang dilakukan.

3.4 Tahap Implementasi

Tahap ini dilakukan implementasi aplikasi berdasarkan perancangan yang telah dibuat sebelumnya. Tahap implementasi perangkat lunak memerlukan beberapa perangkat lunak yang menunjang pembuatan aplikasi sistem. Diharapkan implementasi berjalan sesuai dengan perancangan yang telah dibuat sehingga aplikasi dapat melakukan proses enkripsi dan dekripsi data.

3.5 Tahap Pengujian

Tahap ini dilakukan pengujian setelah aplikasi telah berjalan sesuai dengan perancangan yang telah dibuat. Di dalam tahap ini akan didapatkan data-data yang akan dianalisis untuk digunakan sebagai penilaian kinerja algoritme SPECK dalam mengenkripsi dan dekripsi *file*.

3.6 Tahap Analisis Hasil Pengujian

Tahap ini mendapatkan hasil dari pengujian aplikasi dan melakukan analisis terhadap hasil pengujian untuk menilai kinerja algoritme SPECK. Dari hasil analisis maka dapat dilakukan penarikan kesimpulan dari penelitian yang telah dilakukan.

3.7 Penarikan Kesimpulan dan Saran

Penarikan kesimpulan dilakukan dari hasil analisis yang telah didapatkan dari aplikasi. Kesimpulan merupakan rangkuman seluruh proses analisis kinerja algoritme SPECK. Pemberian saran berguna untuk pengembangan penelitian lebih lanjut dan memperbaiki kesalahan dari penelitian yang telah dibuat.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 4 PERANCANGAN

4.1 Perancangan Implementasi Algoritme

Algoritme SPECK adalah algoritme *block cipher* yang diperkenalkan oleh NSA pada Juni 2013. Algoritme SPECK dalam proses enkripsi dan dekripsi menggunakan kunci yang sama. Terdapat tiga proses utama pada algoritme SPECK yaitu enkripsi, dekripsi dan pembangkitan kunci.

4.1.1 Enkripsi

Enkripsi pada SPECK2n menggunakan operasi di bawah ini pada n-bit *words*:

1. *bitwise* XOR (⊕),

2. *addition modulo* 2^n (+), dan

3. Pergeseran rotasi kiri dan kanan (S, S^-)

Jika nilai rotasi konstanta α = 7 dan β = 2 maka n = 16 (*block size* = 32) dan nilai rotasi α = 8 dan β = 3 untuk sisa varian lainnya. *Pseudocode* proses enkripsi ditunjukkan pada *Pseudocode* 1.

Pseudocode 1: *Pseudocode* Enkripsi

```
Input: X(2n); k0, …, kT-1
Output: Y(2n)
Proses:
1. X0(n) | X1(n)←X(2n)
2. Untuk i = 1 sampai dengan T-1
        X0 ← ((S-α X0) + X1) ⊕ ki
        X1 ← (Sβ X1) ⊕ X0
3. Output:
   Y(2n) ← X0(n) | X1(n)
```

Sumber: Diadaptasi dari Ray Beaulieu et al. (2013)

4.1.2 Dekripsi

Invers dari proses enkripsi adalah untuk menjalankan proses dekripsi menggunakan *modular subtraction* (-). *Pseudocode* proses dekripsi ditunjukkan pada *Pseudocode* 2.

Pseudocode 2: *Pseudocode* Dekripsi

```
Input: Y(2n); k0, …, kT-1
Output: X(2n)
Proses:
1. Y0(n) | Y1(n)←Y(2n)
2. Untuk i = 1 sampai dengan T-1
        Y0 ← Sα((Y0 ⊕ kT-i-1) - Y1)
        Y1 ← S-β (Y0 ⊕ Y1)
3. Output
   X(2n) ← Y0(n) | Y1(n)
```

Sumber: Diadaptasi dari Ray Beaulieu et al. (2013)

4.1.3 Pembangkitan Kunci

Proses pembangkitan kunci memiliki proses yang sama dengan enkripsi tetapi kunci yang digunakan menjadi nilai *round*. *Pseudocode* proses pembangkitan kunci ditunjukkan pada *Pseudocode* 3.

Pseudocode 3: *Pseudocode* Pembangkitan Kunci

```
1  Input: K(mn)
2  Output: k0, …, kT-1
3  Proses:
4  1. k0(n) | l0(n) | … | lm-2(n) ← K(mn)
5  2. Untuk i = 1 sampai dengan T-1
6         tmp ← (ki +(S-α l0 )) ⊕ i
7         ki+1 ← (Sβ ki) ⊕ tmp
8  3. Output
9     K0(n) | k1(n) | … | kT-1(n)
```

Sumber: Diadaptasi dari Ray Beaulieu et al. (2013)

4.1.4 Manualisasi Algoritme SPECK

Pada perancangan implementasi algoritme juga terdapat manualisasi untuk salah satu varian algoritme SPECK yaitu SPECK128/128 yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Manualisasi Algoritme SPECK

Masukan	*Plaintext* 1	456
	Plaintext 2	789
	Key 1	12
	Key 2	34
Key schedule	*Round* 1	Melakukan fungsi *round* dengan x = 18, y = 52, k = 0 • Menghitung nilai x: Konversi 18 ke biner: 00000000000000000000000000000000000000000000000000000000000010010 Rotasi ke kanan 8 kali: 00010010000000000000000000000000000000000000000000000000000000000 x OR y: 00010010000000000000000000000000000000000000000000000000000110100 x XOR k: 00010010000000000000000000000000000000000000000000000000000110100 • Menghitung nilai y: Konversi 52 ke biner: 00000000000000000000000000000000000000000000000000000000000110100 Rotasi ke kiri 3 kali:

Tabel 4.1 Manualisasi Algoritme SPECK (lanjutan)

		00000000000000000000000000000000000000000000000000000000000110100000 y XOR x: 00010010000000000000000000000000000000000000000000000000000110010100 Hasil round: 10010000000000000000000000000000000000000000000000000000000110100, 10010000000000000000000000000000000000000000000000000000000110010100
Enkripsi	*Round* 1	Melakukan fungsi *round* dengan x = 1110, y = 1929, k = 52 • Menghitung nilai x: Konversi 1110 ke biner: 00000000000000000000000000000000000000000000000000000000010001010110 Rotasi ke kanan 8 kali: 01010110000000000000000000000000000000000000000000000000000000000100 x or y: 10101100000000000000000000000000000000000000000000000000000011110001101 x XOR k: 10101100000000000000000000000000000000000000000000000000000011110111001 • Menghitung nilai y: Konversi 1929 ke biner: 00000000000000000000000000000000000000000000000000000000000011110001001 Rotasi ke kiri 3 kali: 00000000000000000000000000000000000000000000000000000000011110001001000 y XOR x: 10101100000000000000000000000000000000000000000000000000011101111110001 Hasil: 10101100000000000000000000000000000000000000000000000000000011110111001, 10101100000000000000000000000000000000000000000000000000011101111110001

4.2 Perancangan Sistem

Pada perancangan sistem terdapat dua tahap yaitu tahap analisis lingkungan sistem di mana terdapat analisis kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam penelitian serta kebutuhan fitur untuk proses dalam sistem. Tahap kedua adalah perancangan aplikasi sistem di mana akan dijelaskan alur kerja aplikasi secara keseluruhan dimulai dari pengguna melakukan *input plaintext* atau *ciphertext* dan kunci sampai dengan dilakukannya proses enkripsi dan dekripsi *file* yang menghasilkan *output* berupa *ciphertext* untuk proses enkripsi dan *plaintext* untuk proses dekripsi.

4.2.1 Lingkungan Sistem

Setelah melakukan studi literatur, tahap selanjutnya adalah analisis lingkungan sistem di mana terdapat kebutuhan perangkat keras dan perangkat lunak sebagai berikut:

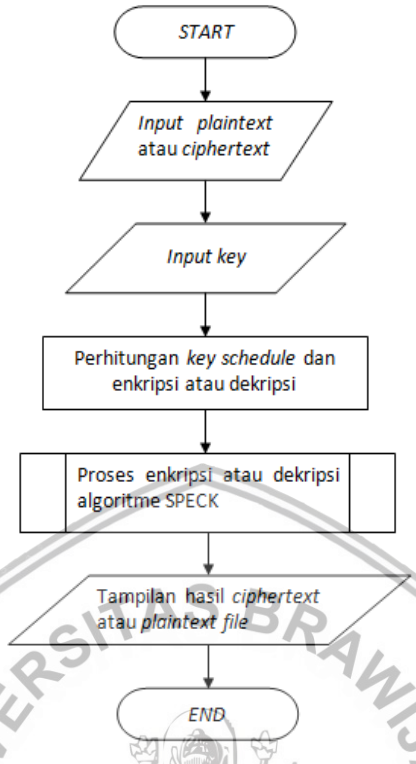
1. Kebutuhan perangkat keras:

   1 unit laptop

2. Kebutuhan perangkat lunak:
   a. Netbeans IDE 8.1
   b. jdk1.8.0_60
3. Kebutuhan Fitur:
   a. Menampilkan *file* dari *driver* pengguna
   b. Mengembalikan kondisi awal sistem
   c. Proses enkripsi dan deskripsi
   d. Simpan hasil proses

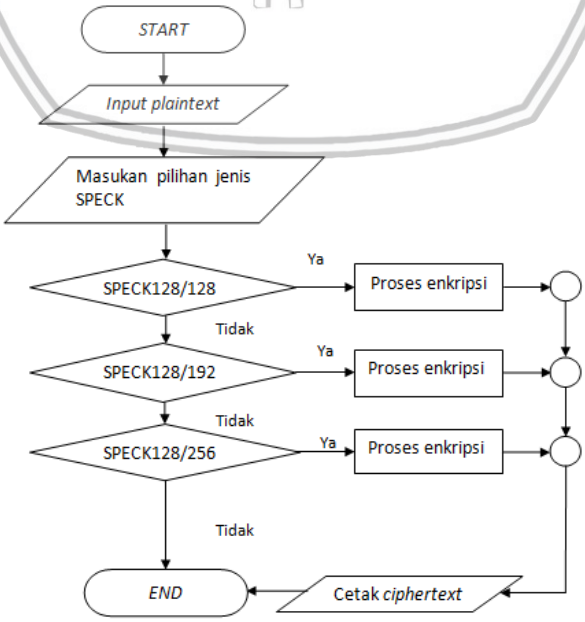
4.2.2 Perancangan Aplikasi Sistem

Tahap perancangan aplikasi digambarkan pada diagram alir aplikasi. Implementasi aplikasi dimulai dari pengguna meng*input*kan *plaintext* atau *ciphertext* kemudian pengguna *input key*. Selanjutnya adalah perhitungan *key schedule* dan enkripsi atau dekripsi. Pada proses perhitungan *key schedule* tercipta sejumlah kunci sesuai dengan jumlah *round* yang ditetapkan berdasarkan varian SPECK yang dipilih dan digunakan untuk perhitungan proses enkripsi dan dekripsi. Kemudian terdapat subproses enkripsi dan dekripsi data dengan menggunakan algoritme SPECK. Dan terakhir keluar hasil *ciphertext* atau *plaintext* data dari masing-masing proses enkripsi dan dekripsi. Berikut adalah diagram aplikasi yang ditunjukkan pada Gambar 4.1.



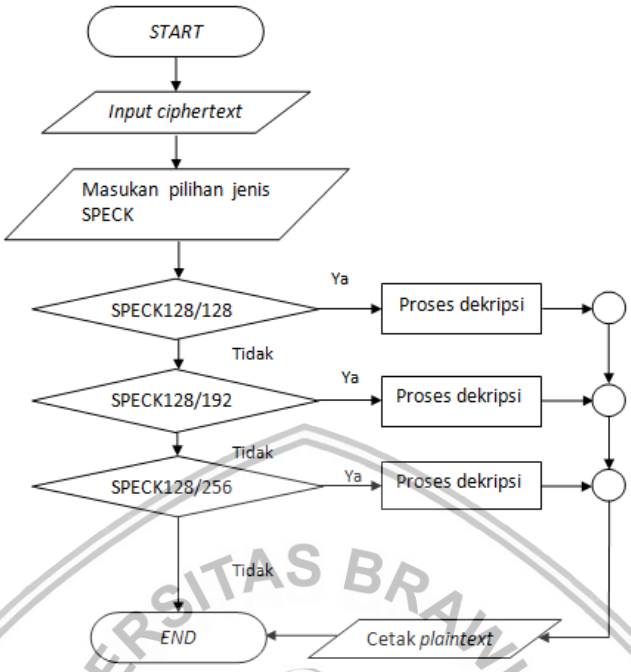
Gambar 4.1 Diagram alir aplikasi

Pada proses enkripsi dan dekripsi data pada aplikasi sistem akan terdapat beberapa pilihan yang mengharuskan pengguna untuk memilih salah satu dari pilihan yang ada yaitu berupa varian algoritme SPECK berdasarkan ukuran blok dan kunci kemudian sistem akan mencetak data. Apabila pengguna tidak memilih salah satu dari ketiga varian SPECK, maka pengguna keluar dari sistem aplikasi. Berikut adalah diagram alir subproses enkripsi data pada Gambar 4.2.



Gambar 4.2 Diagram alir subproses enkripsi data

Diagram alir subproses dekripsi data dapat dilihat pada Gambar 4.3.



Gambar 4.3 Diagram alir subproses dekripsi data

4.3 Perancangan Pengujian

Pada tahap perancangan pengujian akan dijabarkan serangkaian pengujian yang dilakukan untuk menguji kevalidan, kinerja dan tingkat keamanan algoritme SPECK dengan melakukan pengujian *Test Vector*, waktu eksekusi proses enkripsi dan dekripsi serta *avalanche effect*.

4.3.1 Pengujian *Test Vector*

Pengujian *Test Vector* adalah suatu pengujian yang ditetapkan oleh pembuat algoritme SPECK untuk menguji kevalidan algoritme tersebut yang berisi *key, plaintext* dan *ciphertext*. Terdapat tiga data uji untuk masing-masing varian algoritme SPECK yaitu sebagai berikut:

a. SPECK128/128

```
Key            : 0f0e0d0c0b0a0908 0706050403020100
Plaintext      : 6c61766975716520 7469206564616d20
Ciphertext     : a65d985179783265 7860fedf5c570d18
```

b. SPECK128/192

```
Key            : 1716151413121110 0f0e0d0c0b0a0908
                 0706050403020100
Plaintext      : 7261482066656968 43206f7420746e65
Ciphertext     : 1be4cf3a13135566 f9bc185de03c1886
```

c. SPECK128/256

```
Key            : 1f1e1d1c1b1a1918 1716151413121110
                 0f0e0d0c0b0a0908 0706050403020100
Plaintext      : 65736f6874206e49 202e72656e6f6f70
Ciphertext     : 4109010405c0f53e 4eeeb48d9c188f43
```

4.3.2 Pengujian Waktu Eksekusi

Pengujian waktu eksekusi adalah pengujian waktu yang diperlukan algoritme SPECK untuk melakukan proses enkripsi dan dekripsi pada setiap varian algoritme SPECK. Ukuran file yang diuji terdiri dari enam varian yaitu 50 KB, 100 KB, 150 KB, 200 KB, 250 KB dan 300 KB. Setiap pengujian varian algoritme SPECK akan dilakukan sebanyak 1024 kali percobaan pada ukuran file 50 KB dan 100 KB sedangkan pada ukuran file lainnya dilakukan sebanyak 170 kali percobaan. Hal tersebut terjadi karena adanya pengaruh waktu sistem yang dapat dilihat pada sub bab 6.4.

Jumlah percobaan yang dilakukan bertujuan untuk mengetahui kinerja algoritme SPECK. Selain kinerja algoritme SPECK, terdapat masukan variasi *key* yang digunakan berdasarkan urutan biner yang juga diparameteri oleh *key size* pada masing-masing SPECK sehingga mendapatkan variasi biner yang dapat mencapai 1024 variasi kunci. Hasil dari catatan waktu yang didapat akan dianalisis untuk menilai kinerja algoritme SPECK dengan melakukan pengujian Kolmogorov-Smirnov, Kruskal Wallis dan *Post Hoc test*.

4.3.3 Pengujian *Avalanche Effect*

Pengujian *Avalanche Effect* adalah pengujian tingkat keamanan algoritme kriptografi yang dapat dikatakan baik apabila perubahan pada 1 bit *input* berupa *plaintext* maupun *key* akan berpengaruh besar pada hasil *output* (Ariyanto, Pravitasari dan Setyorini, 2008). Dalam penelitian akan dilakukan perubahan masing-masing pada *plaintext* dan *key* untuk masing-masing varian algoritme SPECK.

BAB 5 IMPLEMENTASI

Bab ini menjelaskan tahapan dalam mengimplementasikan perancangan yang telah dibuat menggunakan algoritme SPECK. Implementasi terdiri dari fungsi-fungsi untuk menjalankan proses enkripsi, dekripsi dan *key schedule* untuk format *file* PDF. Implementasi dilakukan pada tiga variasi SPECK yaitu SPECK128/128, SPECK128/192 dan SPECK128/256. *Source code* 1 menampilkan inisialisasi variabel dalam proses algoritme SPECK.

Source code 1: Inisialisasi variabel algoritme SPECK

```
1  private long[] k;
2  long x;
3  private long y;
```

5.1 Implementasi Fungsi Mask

Fungsi mask bertujuan untuk melakukan masker pada semua bit yang lebih tinggi dari ukuran words cipher ini dalam nilai yang disediakan. Variabel val adalah nilai untuk mask. Fungsi mask ditunjukkan pada *Source code* 2.

Source code 2: Fungsi Mask

```
1  protected long mask(long val) {
2          return val;
3      }
```

5.2 Implementasi Fungsi rotl

Fungsi rotl bertujuan untuk melakukan pergeseran putaran ke kiri sebanyak nilai yang ditentukan dan fungsi ini digunakan pada proses enkripsi, dekripsi dan *key schedule*. Fungsi rotl ditunjukkan pada *Source code* 3.

Source code 3: Fungsi rotl

```
1  private long rotl(long i, int distance) {
2          return (i << distance) | (i >>> (wordSizeBits - distance));
3      }
```

5.3 Implementasi Fungsi rotr

Fungsi rotr bertujuan untuk melakukan pergeseran putaran ke kanan sebanyak nilai yang ditentukan dan fungsi ini digunakan pada proses enkripsi, dekripsi dan *key schedule*. Fungsi rotr ditunjukkan pada *Source code* 4.

Source code 4: Fungsi rotr

```
1  private long rotr(long i, int distance) {
2          return (i >>> distance) | (i << (wordSizeBits - distance));
3      }
```

5.4 Implementasi Merubah Byte ke Word

Implementasi ini bertujuan untuk merubah byte dari data *input* ke dalam bentuk word dan digunakan pada proses *key schedule*. Implementasi merubah byte ke word ditunjukkan pada *Source code* 5.

Source code 5: Fungsi Merubah Byte ke Word

```
1   private long bytesToWord(final byte[] bytes, final int off) {
2           if ((off + wordSize) > bytes.length) {
3               throw new IllegalArgumentException();
4           }
5           long word = 0;
6           int index = off;
7           word = (bytes[index++] & 0xffl);
8           word = (word << 8) | (bytes[index++] & 0xffl);
9           word = (word << 8) | (bytes[index++] & 0xffl);
10          word = (word << 8) | (bytes[index++] & 0xffl);
11          word = (word << 8) | (bytes[index++] & 0xffl);
12          word = (word << 8) | (bytes[index++] & 0xffl);
13          if (wordSize == 8) {
14              word = (word << 8) | (bytes[index++] & 0xffl);
15              word = (word << 8) | (bytes[index++] & 0xffl);
16          }
17          return word;
18      }
```

5.5 Implementasi *Key Schedule*

Implementasi *key schedule* bertujuan untuk menciptakan kunci di setiap *round* yang telah ditentukan dan digunakan untuk melakukan proses enkripsi dan dekripsi. Proses *key schedule* memanggil fungsi mask, rotl, rotr dan perubahan byte ke word. Implementasi *key schedule* ditunjukkan pada *Source code* 6.

Source code 6: Fungsi *Key Schedule*

```
1   protected void KeySchedule(byte[] keyBytes) {
2           // Tentukan jumlah words kunci m
3           int keyWords = keyBytes.length / wordSize;
4           // Jumlah putaran ditambah 1 untuk setiap words kunci> 2
5           rounds = baseRounds + (keyWords - 2);
6           k = new int[rounds];
7           // Load k[0]
8           k[0] = bytesToWord(keyBytes, (keyWords - 1) * wordSize);
9           // Load l [m-2] ... l [0], sisakan ruang untuk l [m-1] dalam
10  ekspansi kunci
11          final long[] l = new long[keyWords];
12          for (int i = 0; i < keyWords - 1; i++) {
13              l[i] = bytesToWord(keyBytes, (keyWords - i - 2) *
14  wordSize);
15          }
16
17          // ekspansi kunci menggunakan fungsi bulat di atas l [m-2]
18  ... l [0], k [0] dengan angka bulat sebagai kunci
19          for (int i = 0; i < rounds - 1; i++) {
20              final int lw = (i + keyWords - 1) % keyWords;
21              l[lw] = mask((rotr(l[i % keyWords], alpha) + k[i]) ^ i);
22              k[i + 1] = mask(rotl(k[i], beta) ^ l[lw]);
23          }
24      }
```

5.6 Implementasi Proses Enkripsi Algoritme SPECK

Implementasi proses enkripsi algoritme SPECK ditampilkan dalam fungsi *encryptBlock*(). Fungsi ini digunakan untuk mengubah *plaintext* menjadi *ciphertext*. Proses enkripsi memanggil fungsi mask, rotr dan rotl. Potongan kode program untuk implementasi proses enkripsi ditunjukkan pada *Source code* 7.

Source code 7: Fungsi Enkripsi

```
1   protected void encryptBlock() {
2           int x = this.x;
3           int y = this.y;
4
5           for (int r = 0; r < rounds; r++) {
6               x = mask((rotr(x, alpha) + y) ^ k[r]);
7               y = mask(rotl(y, beta) ^ x);
8           }
9           this.x = x;
10          this.y = y; }
```

5.7 Implementasi Proses Dekripsi Algoritme SPECK

Implementasi proses dekripsi algoritme SPECK ditampilkan dalam fungsi *decryptBlock*(). Fungsi ini digunakan untuk mengubah *ciphertext* menjadi *plaintext*. Proses dekripsi memanggil fungsi mask, rotr dan rotl. Potongan kode program untuk implementasi proses dekripsi ditunjukkan pada *Source code* 8.

Source code 8: Fungsi Dekripsi

```
1   protected void decryptBlock() {
2           int x = this.x;
3           int y = this.y;
4
5           for (int r = rounds - 1; r >= 0; r--) {
6               y = mask(rotr(x ^ y, beta));
7               x = mask(rotl(mask((x ^ k[r]) - y), alpha));
8           }
9           this.x = x;
10          this.y = y;
11      }
```

5.8 Implementasi Proses Enkripsi dalam Sistem

Implementasi proses enkripsi dalam sistem terdapat pada fitur proses dengan menciptakan fungsi btnProsesActionPerformed(). *Line* pertama menunjukkan pengaturan waktu mulai proses enkripsi. *Line* 3 menunjukkan proses enkripsi di*looping* sebanyak masukan *byte* data. *Line* 4-6 untuk menyediakan slot karakter sesuai dengan variasi SPECK. *Line* 9-11 adalah kondisi jika proses melebihi slot karakter yang disediakan. *Line* 13-14 adalah kondisi jika proses sesuai dengan jumlah slot karakter yang disediakan. *Line* 16 memanggil fungsi enkripsi pada *class Main* dan melakukan proses enkripsi kemudian disimpan dalam variabel byteResult. *Line* 21-25 untuk memasukan hasil enkripsi dalam bentuk byte ke variabel sbHasil. *Line* 28-29 melakukan pengaturan waktu berakhir proses enkripsi dan mencatat waktu proses enkripsi keseluruhan yang dihasilkan. Potongan kode program untuk implementasi proses enkripsi dalam sistem ditunjukkan pada *Source code* 9.

Source code 9: Implementasi Proses Enkripsi dalam Sistem

```
1   startTime = System.currentTimeMillis();
2
3                   while (idx < byteData.length) {
4                       byteInput = new byte[maxText];
5                       for (int i = 0; i < byteInput.length; i++) {
6                           byteInput[i] = 0;
7                       }
8
```


repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

```
                    if (idx + maxText > byteData.length) {
                        System.arraycopy(byteData, idx, byteInput, 0,
byteData.length - idx);
                    } else {
                        System.arraycopy(byteData, idx, byteInput, 0,
maxText);
                    }
                    byteResult = prosesEncrypt(Tipes, byteInput,
byteKeys);

                    idx += maxText;

                    byte[] tmpRes = new byte[byteResult.length];
                    for (int i = 0; i < byteResult.length; i++) {
                        char c = (char) byteResult[i];
                        tmpRes[i] = (byte) c;
                        sbHasil.append(c);
                    }
                }
                endTime = System.currentTimeMillis();
                duration = endTime - startTime;
```

5.9 Implementasi Proses Dekripsi dalam Sistem

Implementasi proses enkripsi dalam sistem terdapat pada fitur proses dengan menciptakan fungsi btnProsesActionPerformed(). Proses dekripsi dalam sistem memiliki proses yang sama dengan enkripsi dan hanya dibedakan pada *line* 18 yaitu pemanggilan fungsi dekripsi pada *class main* dan melakukan proses dekripsi kemudian disimpan ke dalam variabel byteResult. Potongan kode program untuk implementasi proses enkripsi dalam sistem ditunjukkan pada *Source code* 10.

Source code 10: Implementasi Proses Dekripsi dalam Sistem

```
startTime = System.currentTimeMillis();

            idx = 0;
            while (idx < byteData.length) {
                byteInput = new byte[maxText];
                for (int i = 0; i < byteInput.length; i++) {
                    byteInput[i] = 0;
                }

                if (idx + maxText > byteData.length) {
                    System.arraycopy(byteData, idx, byteInput, 0,
byteData.length - idx);
                } else {
                    System.arraycopy(byteData, idx, byteInput, 0,
maxText);
                }

                byteResult = prosesDecrypt(Tipes, byteInput,
byteKeys);
                idx += maxText;
                byte[] tmpRes = new byte[byteResult.length];
                for (int i = 0; i < byteResult.length; i++) {
                    char c = (char) byteResult[i];
                    tmpRes[i] = (byte) c;
                    sbHasil.append(c);
                }
            }
            endTime = System.currentTimeMillis();
            duration = endTime - startTime;
```


repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

5.10 Implementasi Proses *File* PDF

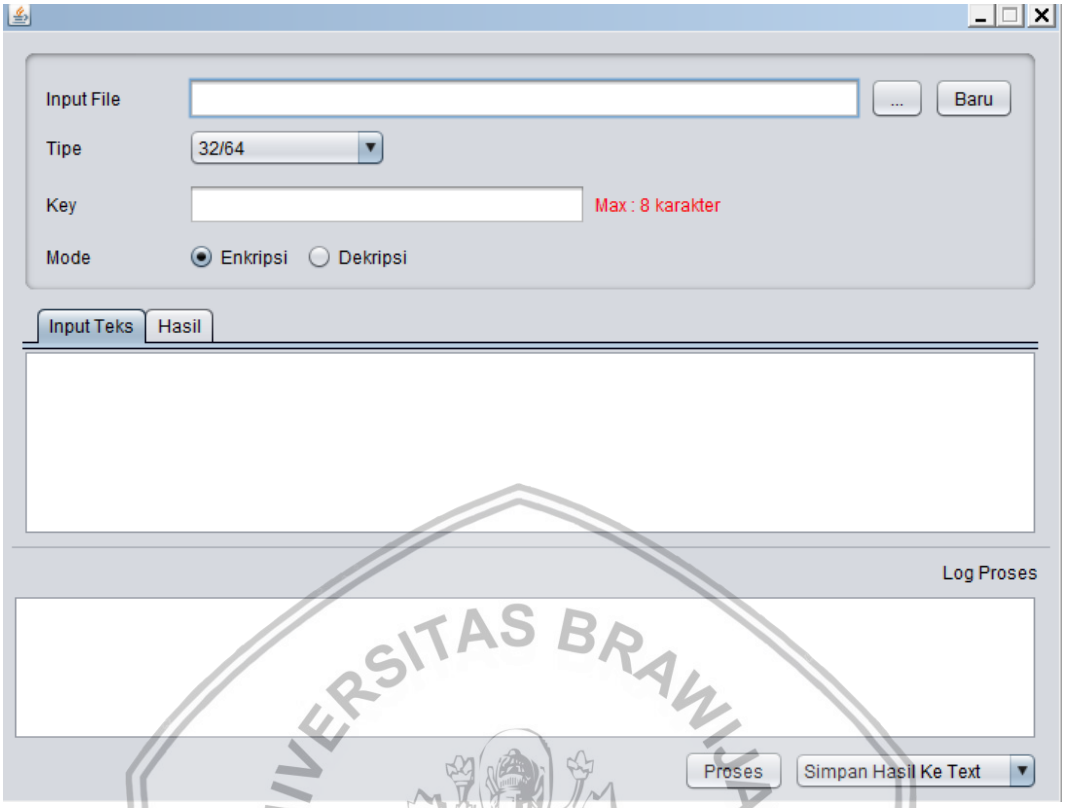
Implementasi proses *file* PDF dengan melakukan *load* data PDF dalam sistem yang kemudian akan disimpan dalam variabel sbHasil. Pemanggilan variabel sbHasil saat menyimpan hasil *ciphertext* dan *plaintext*. Potongan kode program untuk implementasi proses *file* PDF dalam sistem ditunjukkan pada *Source code* 11.

Source code 11: Implementasi Proses *File PDF*

```
try {

                byteData =
Files.readAllBytes(Paths.get(txtLokasi.getText().trim()));
                if (txtLokasi.getText().trim().endsWith(".pdf")) {
                    PDDocument document =
PDDocument.load(txtLokasi.getText().trim());
                    document.getClass();
                    if (document.isEncrypted()) {
                        try {
                            document.decrypt("");
                        } catch (Exception e) {
                            System.err.println("Error:  " +
e.toString());
                            System.exit(1);
                        }
                    }

                    PDFTextStripper stripper = new PDFTextStripper();
                    stripper.setSortByPosition(true);
                    stripper = new PDFTextStripper();
                    String s = stripper.getText(document);
                    String[] arrStr = s.split(" ");
                    byteData = new byte[arrStr.length - 1];
                    for (int i = 0; i < arrStr.length; i++) {
                        if (arrStr[i].trim().length() == 0) {
                            continue;
                        }
                        byte b = (byte)
Integer.parseInt(arrStr[i].trim());
                        //System.out.println( b);
                        byteData[i] = b;
                    }
                    document.close();;
                }
            } catch (IOException ex) {

Logger.getLogger(FrmMain.class.getName()).log(Level.SEVERE, null,
ex);
            }

            sbHasil = new StringBuffer();
```

5.11 Tampilan *Interface* Sistem

Tampilan *interface* sistem saat semua fungsi digabung membentuk sebuah sistem yang dapat melakukan proses enkripsi dan dekripsi *file* teks menggunakan algortime SPECK dengan memanfaatkan fitur utama *upload*, baru, proses dan simpan yang ditunjukkan pada Gambar 5.1.



Gambar 5.1 Tampilan *interface* sistem

BAB 6 PENGUJIAN

Bab ini akan menjelaskan pengujian yang dilakukan berdasarkan perancangan yang telah dibuat sebelumnya. Terdapat tiga tahap pengujian yaitu pengujian *test vector* untuk menguji ke*valid*an algoritme SPECK, pengujian waktu eksekusi di mana akan menghasilkan performansi algoritme SPECK dan pengujian *Avalanche Effect* untuk menguji tingkat keamanan algoritme SPECK.

6.1 Pengujian *Test Vector*

Pengujian *test vector* digunakan untuk menguji ke*valid*an suatu algoritme di mana *test vector* telah ditetapkan oleh pembuat algoritme SPECK berupa *input plaintext*, *key* dan *ciphertext*.

6.1.1 Pengujian *Test Vector* SPECK128/128

Hasil proses enkripsi dan dekripsi dari sistem ditunjukkan pada Gambar 6.1.

```
-----Speck128/128-----
key :
0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00
text :
6C 61 76 69 75 71 65 20 74 69 20 65 64 61 6D 20
==========>> Hasil Enkripsi :
A6 5D 98 51 79 78 32 65 78 60 FE DF 5C 57 0D 18
```

Gambar 6.1 Hasil *test vector* SPECK128/128

Test vector dalam *paper*:

```
Key        : 0f0e0d0c0b0a0908 0706050403020100
Plaintext  : 6c61766975716520 7469206564616d20
Ciphertext : a65d985179783265 7860fedf5c570d18
```

6.1.2 Pengujian *Test Vector* SPECK128/192

Hasil proses enkripsi dan dekripsi dari sistem ditunjukkan pada Gambar 6.2.

```
-----Speck128/192-----
key :
17 16 15 14 13 12 11 10 0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00
text :
72 61 48 20 66 65 69 68 43 20 6F 74 20 74 6E 65
==========>> Hasil Enkripsi :
1B E4 CF 3A 13 13 55 66 F9 BC 18 5D E0 3C 18 86
```

Gambar 6.2 Hasil *test vector* SPECK128/192

Test vector dalam *paper*:

```
Key        : 1716151413121110 0f0e0d0c0b0a0908 0706050403020100
Plaintext  : 7261482066656968 43206f7420746e65
Ciphertext : 1be4cf3a13135566 f9bc185de03c1886
```

6.1.3 Pengujian *Test Vector* SPECK128/256

Hasil proses enkripsi dan dekripsi dari sistem ditunjukkan pada Gambar 6.3.

```
-----Speck128/256-----
key :
1F 1E 1D 1C 1B 1A 19 18 17 16 15 14 13 12 11 10 0F 0E 0D 0C 0B 0A 09 08 07 06 05 04 03 02 01 00
text :
65 73 6F 68 74 20 6E 49 20 2E 72 65 6E 6F 6F 70
 ====================>> Hasil Enkripsi :
 41 09 01 04 05 C0 F5 3E 4E EE B4 8D 9C 18 8F 43
```

Gambar 6.3 Hasil *test vector* SPECK128/256

Test vector dalam *paper*:

```
Key         :1f1e1d1c1b1a1918   1716151413121110   0f0e0d0c0b0a0908
             0706050403020100
Plaintext   : 65736f6874206e49 202e72656e6f6f70
Ciphertext  : 4109010405c0f53e 4eeeb48d9c188f43
```

6.2 Pengujian Waktu Eksekusi

Pengujian waktu eksekusi proses enkripsi dan dekripsi dilakukan menggunakan 6 varian ukuran *file* yaitu 50 KB, 100 KB, 150 KB, 200 KB, 250 KB dan 300 KB untuk masing-masing varian algoritme SPECK. Pada file ukuran 50 KB dan 100 KB diuji sebanyak 1024 kali percobaan dan file ukuran 150 KB, 200 KB, 250 KB dan 300 KB diuji sebanyak 170 kali percobaan. Dalam hal ini, varian SPECK yang akan diuji adalah SPECK128/128, SPECK128/192 dan SPECK128/256. Hasil catatan waktu enkripsi dan dekripsi yang dihasilkan oleh sistem ditunjukkan pada Tabel 6.1 – 6.6.

Tabel 6.1 Hasil uji waktu enkripsi dan dekripsi *file* 50 KB

	SPECK128/128 dengan satuan waktu (ms)		SPECK128/192 dengan satuan waktu (ms)		SPECK128/256 dengan satuan waktu (ms)	
No.	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi
1	1000	1082	980	1002	950	990
2	990	1050	920	972	1000	954
3	930	920	970	1000	930	970
4	1020	1030	880	982	880	912
5	940	1040	870	960	920	960
6	910	944	900	920	1030	860
7	990	930	1010	860	970	870
.....						
1018	920	1060	1030	920	980	950
1019	930	930	890	870	880	1020
1020	990	940	1010	1010	950	850
1021	940	900	880	960	930	1000
1022	950	1030	980	860	880	910
1023	990	950	880	920	970	970
1024	930	920	960	860	920	870

Tabel 6.2 Hasil uji waktu enkripsi dan dekripsi *file* 100 KB

	SPECK128/128 dengan satuan waktu (ms)		SPECK128/192 dengan satuan waktu (ms)		SPECK128/256 dengan satuan waktu (ms)	
No.	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi
1	2400	2402	2302	2460	2360	2422
2	2460	2440	2410	2300	2290	2430
3	2500	2450	2400	2310	2340	2310
4	2440	2370	2340	2320	2300	2300
5	2390	2330	2370	2280	2270	2390
6	2300	2300	2410	2260	2300	2350
7	2470	2330	2400	2330	2370	2280
.....						
1018	2340	2320	2410	2380	2460	2430
1019	2430	2320	2290	2360	2380	2460
1020	2410	2390	2300	2250	2290	2300
1021	2350	2400	2340	2340	2350	2520
1022	2450	2470	2360	2290	2300	2280
1023	2500	2340	2520	2370	2410	2450
1024	2390	2380	2500	2520	2440	2480

Tabel 6.3 Hasil uji waktu enkripsi dan dekripsi *file* 150 KB

	SPECK128/128 dengan satuan waktu (ms)		SPECK128/192 dengan satuan waktu (ms)		SPECK128/256 dengan satuan waktu (ms)	
No.	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi
1	4050	3980	3972	3936	3970	3960
2	3890	3850	4070	3910	3900	3800
3	3930	3810	3900	3850	3850	3940
4	3980	4000	3850	3970	3920	3900
5	3850	3990	3970	3840	3900	3810
6	3942	3970	3910	3890	3910	3890
7	3930	3980	3930	3930	3810	3874
.....						
164	3890	4010	3890	3990	3840	3850
165	4080	3800	3850	3992	3890	3910
166	3890	4000	3992	3850	3881	3860
167	3890	3850	3970	3910	3900	3800
168	3942	4000	3970	3840	3910	3810
169	3930	3980	3860	3982	3880	3932
170	4020	3970	3870	3970	3890	3886

Tabel 6.4 Hasil uji waktu enkripsi dan dekripsi *file* 200 KB

	SPECK128/128 dengan satuan waktu (ms)		SPECK128/192 dengan satuan waktu (ms)		SPECK128/256 dengan satuan waktu (ms)	
No.	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi
1	5528	5530	5594	5431	5510	5410
2	5518	5550	5540	5412	5550	5360
3	5512	5560	5580	5454	5440	5310
4	5522	5570	5594	5406	5420	5362
5	5594	5540	5500	5506	5350	5522
6	5540	5610	5420	5460	5410	5350
7	5470	5490	5490	5477	5480	5302
.....						
164	5470	5490	5477	5490	5480	5302
165	5480	5480	5510	5410	5450	5300
166	5590	5480	5506	5530	5340	5300
167	5580	5560	5450	5473	5420	5500
168	5530	5490	5421	5450	5360	5500
169	5594	5590	5360	5570	5380	5460
170	5470	5500	5442	5420	5520	5410

Tabel 6.5 Hasil uji waktu enkripsi dan dekripsi *file* 250 KB

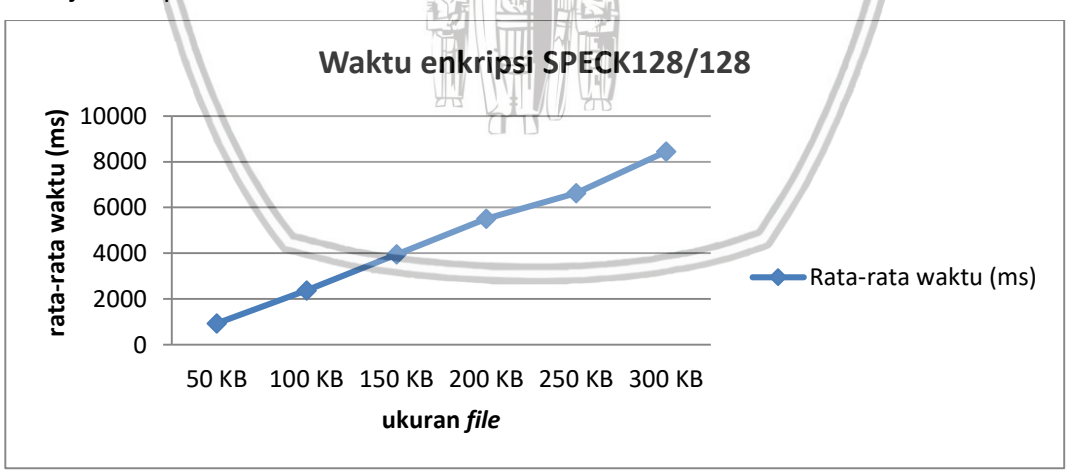
	SPECK128/128 dengan satuan waktu (ms)		SPECK128/192 dengan satuan waktu (ms)		SPECK128/256 dengan satuan waktu (ms)	
No.	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi
1	6720	6774	6582	6740	6780	6620
2	6750	6650	6630	6640	6540	6680
3	6620	6811	6670	6720	6530	6860
4	6630	6830	6700	6620	6570	6590
5	6640	6770	6590	6810	6670	6610
6	6600	6670	6630	6740	6650	6680
7	6540	6660	6570	6672	6730	6580
.....						
164	6590	6770	6640	6710	6580	6838
165	6570	6640	6740	6740	6644	6580
166	6680	6660	6570	6700	6581	6702
167	6750	6650	6630	6640	6540	6680
168	6670	6770	6590	6740	6670	6610
169	6640	6640	6645	6720	6640	6524
170	6650	6832	6530	6770	6540	6720

Tabel 6.6 Hasil uji waktu enkripsi dan dekripsi *file* 300 KB

	SPECK128/128 dengan satuan waktu (ms)		SPECK128/192 dengan satuan waktu (ms)		SPECK128/256 dengan satuan waktu (ms)	
No.	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi	Waktu Enkripsi	Waktu Dekripsi
1	8430	8472	8400	8390	8330	8486
2	8580	8370	8340	8410	8550	8580
3	8530	8440	8370	8400	8300	8520
4	8480	8482	8390	8460	8410	8370
5	8502	8340	8420	8494	8390	8380
6	8450	8360	8450	8390	8340	8470
7	8470	8522	8440	8480	8542	8490
.....						
164	8320	8540	8450	8350	8410	8350
165	8390	8460	8540	8390	8380	8372
166	8540	8510	8412	8460	8490	8430
167	8532	8540	8410	8380	8520	8420
168	8450	8440	8520	8390	8390	8470
169	8542	8490	8422	8500	8360	8510
170	8422	8380	8420	8440	8440	8450

6.2.1 Pengujian Waktu Enkripsi SPECK128/128

Hasil pengujian waktu enkripsi SPECK128/128 dengan varian ukuran *file* ditunjukkan pada Gambar 6.4.



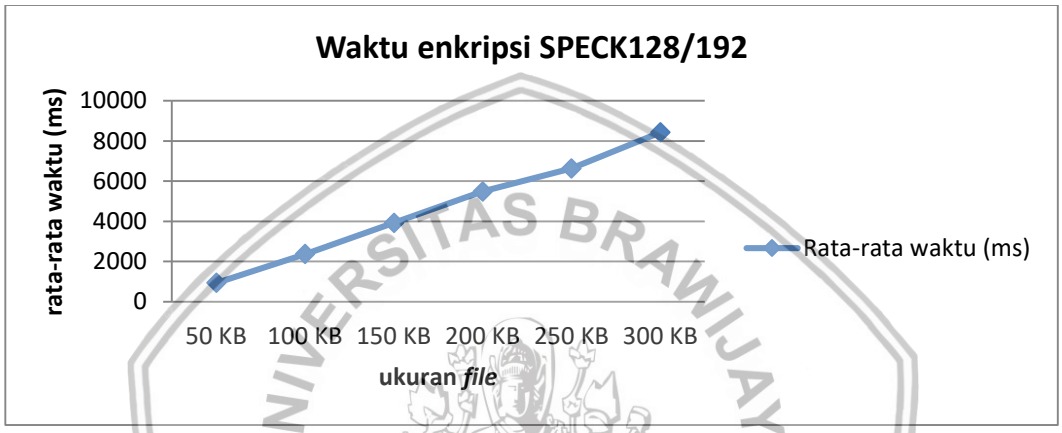
Gambar 6.4 Grafik waktu enkripsi Algoritme SPECK128/128

Pada Gambar 6.4 menunjukkan perbedaan rata-rata waktu enkripsi pada setiap varian ukuran *file*. Pada kiri grafik merupakan keterangan rata-rata waktu enkripsi dan pada bawah grafik merupakan keterangan varian ukuran *file*. Rata-rata waktu yang dihasilkan *file* 50 KB adalah 932,82 ms, *file* 100 KB adalah 2375,15 ms, *file* 150 KB adalah 3948,77 ms, *file* 200 KB adalah 5505,29 ms, *file* 250 KB adalah 6628,72 ms dan *file* 300 KB adalah 8434,26 ms. Berdasarkan grafik yang dihasilkan didapatkan kesimpulan bahwa pada SPECK128/128 semakin

besar ukuran *file* yang digunakan maka waktu untuk sistem melakukan proses enkripsi semakin lama. Hal ini disebabkan oleh perbedaan panjang kata dan kalimat yang ada pada masing-masing *file* PDF. Pada *file* PDF dengan ukuran 50 KB menghasilkan panjang kata dan kalimat terpendek dibanding ukuran *file* PDF lainnya. Semakin panjang kata dan kalimat maka semakin panjang proses yang dikerjakan SPECK untuk melakukan enkripsi.

6.2.2 Pengujian Waktu Enkripsi SPECK128/192

Hasil pengujian waktu enkripsi SPECK128/192 dengan varian ukuran *file* ditunjukkan pada Gambar 6.5.

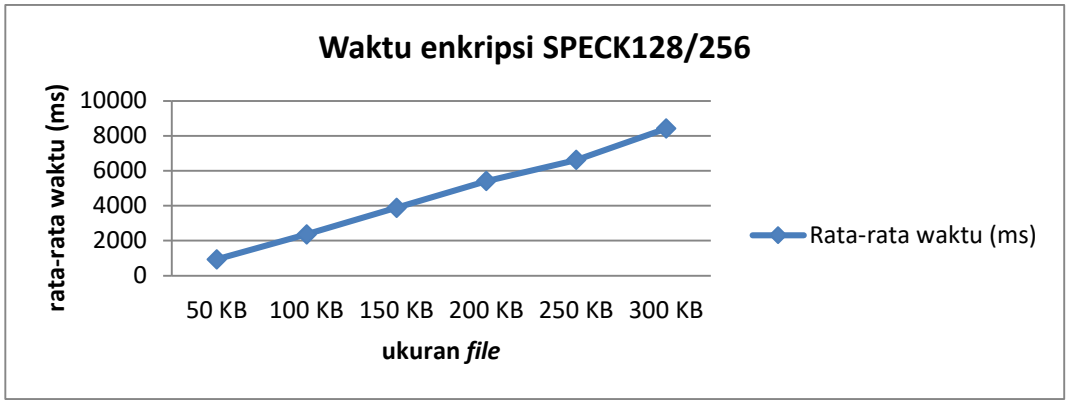


Gambar 6.5 Grafik waktu enkripsi Algoritme SPECK128/192

Pada Gambar 6.5 menunjukkan perbedaan rata-rata waktu enkripsi pada setiap varian ukuran *file*. Pada kiri grafik merupakan keterangan rata-rata waktu enkripsi dan pada bawah grafik merupakan keterangan varian ukuran *file*. Rata-rata waktu yang dihasilkan *file* 50 KB adalah 929,65 ms, *file* 100 KB adalah 2361,71 ms, *file* 150 KB adalah 3921,74 ms, *file* 200 KB adalah 5483,76 ms, *file* 250 KB adalah 6625,73 ms dan *file* 300 KB adalah 8421,73 ms. Berdasarkan grafik yang dihasilkan didapatkan kesimpulan bahwa pada SPECK128/192 semakin besar ukuran *file* yang digunakan maka waktu untuk sistem melakukan proses enkripsi semakin lama. Hal ini disebabkan oleh perbedaan panjang kata dan kalimat yang ada pada masing-masing *file* PDF. Pada *file* PDF dengan ukuran 50 KB menghasilkan panjang kata dan kalimat terpendek dibanding ukuran *file* PDF lainnya. Semakin panjang kata dan kalimat maka semakin panjang proses yang dikerjakan SPECK untuk melakukan enkripsi.

6.2.3 Pengujian Waktu Enkripsi SPECK128/256

Hasil pengujian waktu enkripsi SPECK128/256 dengan varian ukuran *file* ditunjukkan pada Gambar 6.6.

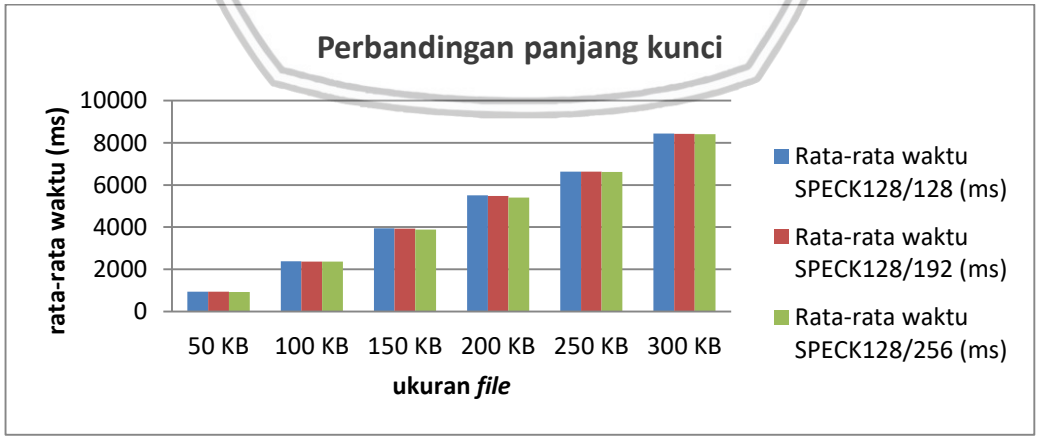


Gambar 6.6 Grafik waktu enkripsi Algoritme SPECK128/256

Pada Gambar 6.6 menunjukkan perbedaan rata-rata waktu enkripsi pada setiap varian ukuran *file*. Pada kiri grafik merupakan keterangan rata-rata waktu enkripsi dan pada bawah grafik merupakan keterangan varian ukuran *file*. Rata-rata waktu yang dihasilkan *file* 50 KB adalah 927,13 ms, *file* 100 KB adalah 2356,69 ms, *file* 150 KB adalah 3879,26 ms, *file* 200 KB adalah 5406,39 ms, *file* 250 KB adalah 6620,41 ms dan *file* 300 KB adalah 8411,07 ms. Berdasarkan grafik yang dihasilkan didapatkan kesimpulan bahwa pada SPECK128/256 semakin besar ukuran *file* yang digunakan maka waktu untuk sistem melakukan proses enkripsi semakin lama. Hal ini disebabkan oleh perbedaan panjang kata dan kalimat yang ada pada masing-masing *file* PDF. Pada *file* PDF dengan ukuran 50 KB menghasilkan panjang kata dan kalimat terpendek dibanding ukuran *file* PDF lainnya. Semakin panjang kata dan kalimat maka semakin panjang proses yang dikerjakan SPECK untuk melakukan enkripsi.

6.2.4 Perbandingan Panjang Kunci SPECK Waktu Enkripsi

Hasil pengujian perbandingan panjang kunci SPECK berdasarkan waktu enkripsi ditunjukkan pada Gambar 6.7.



Gambar 6.7 Grafik perbandingan panjang kunci SPECK waktu enkripsi

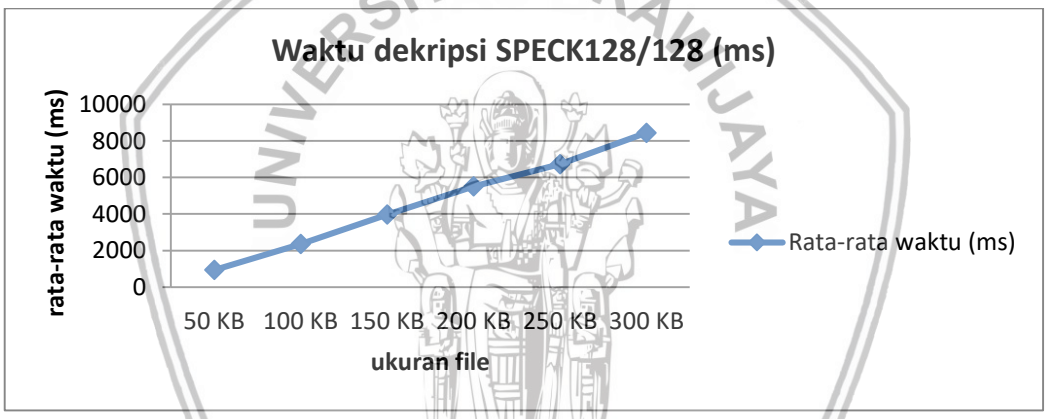
Pada Gambar 6.7 menunjukkan perbandingan rata-rata waktu enkripsi pada setiap varian ukuran *file* berdasarkan panjang kunci yang digunakan. Pada kiri grafik merupakan keterangan rata-rata waktu enkripsi dan pada bawah grafik

merupakan keterangan varian ukuran *file*. Berdasarkan grafik yang dihasilkan, SPECK128/256 menghasilkan rata-rata waktu enkripsi tercepat sedangkan SPECK128/128 menghasilkan rata-rata waktu enkripsi paling lama. Kemudian dapat disimpulkan bahwa semakin panjang kunci yang digunakan maka waktu untuk sistem melakukan proses enkripsi semakin cepat.

Hal diatas disebabkan karena sistem membaca masukan dari pengguna berupa karakter di mana semisal 128 bit menghasilkan sejumlah 16 karakter sedangkan 256 bit menghasilkan sejumlah 32 karakter. SPECK yang mengambil ukuran data 256 bit akan menjalankan 32 karakter dalam sekali proses. Hal ini menyebabkan waktu proses pada ukuran 256 bit menjadi lebih cepat dibandingkan ukuran 128 bit di mana hal tersebut juga menyebabkan SPECK128/256 memiliki waktu proses yang lebih cepat dibanding varian lainnya.

6.2.5 Pengujian Waktu Dekripsi SPECK128/128

Hasil pengujian waktu dekripsi SPECK128/128 dengan varian ukuran *file* ditunjukkan pada Gambar 6.8.

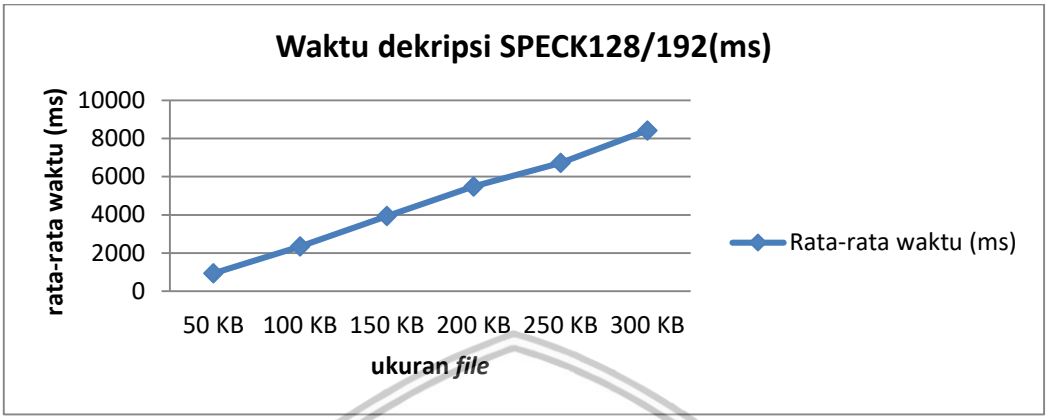


Gambar 6.8 Grafik waktu dekripsi Algoritme SPECK128/128

Pada Gambar 6.8 menunjukkan perbedaan rata-rata waktu dekripsi pada setiap varian ukuran *file*. Pada kiri grafik merupakan keterangan rata-rata waktu dekripsi dan pada bawah grafik merupakan keterangan varian ukuran *file*. Rata-rata waktu yang dihasilkan *file* 50 KB adalah 938,96 ms, *file* 100 KB adalah 2351,62 ms, *file* 150 KB adalah 3962,95 ms, *file* 200 KB adalah 5509,13 ms, *file* 250 KB adalah 6724,17 ms dan *file* 300 KB adalah 8440,21 ms. Berdasarkan grafik yang dihasilkan didapatkan kesimpulan bahwa pada SPECK128/128 semakin besar ukuran *file* yang digunakan maka waktu untuk sistem melakukan proses dekripsi semakin lama. Hal ini disebabkan oleh perbedaan panjang kata dan kalimat yang ada pada masing-masing *file* PDF. Pada *file* PDF dengan ukuran 50 KB menghasilkan panjang kata dan kalimat terpendek dibanding ukuran *file* PDF lainnya. Semakin panjang kata dan kalimat maka semakin panjang proses yang dikerjakan SPECK untuk melakukan dekripsi.

6.2.6 Pengujian Waktu Dekripsi SPECK128/192

Hasil pengujian waktu dekripsi SPECK128/192 dengan varian ukuran *file* ditunjukkan pada Gambar 6.9.

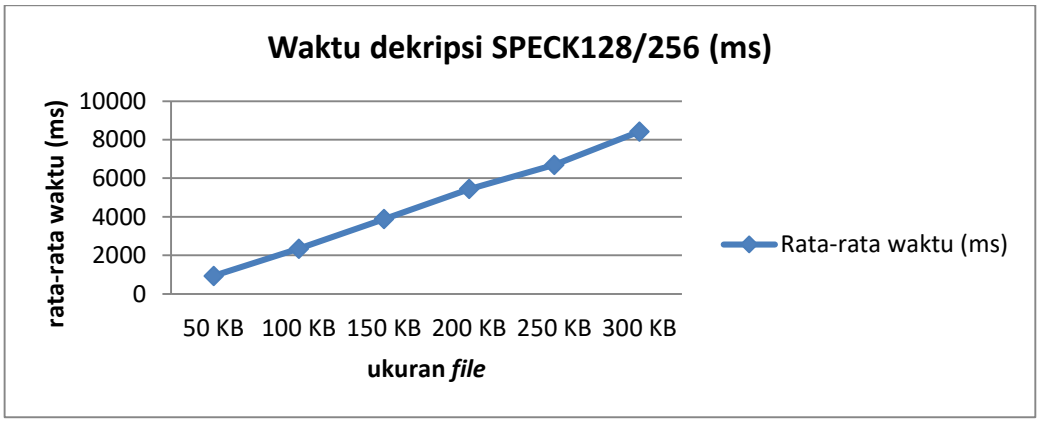


Gambar 6.9 Grafik waktu dekripsi Algoritme SPECK128/192

Pada Gambar 6.9 menunjukkan perbedaan rata-rata waktu dekripsi pada setiap varian ukuran *file*. Pada kiri grafik merupakan keterangan rata-rata waktu dekripsi dan pada bawah grafik merupakan keterangan varian ukuran *file*. Rata-rata waktu yang dihasilkan *file* 50 KB adalah 931,69 ms, *file* 100 KB adalah 2343,70 ms, *file* 150 KB adalah 3938,47 ms, *file* 200 KB adalah 5491,81 ms, *file* 250 KB adalah 6720,88 ms dan *file* 300 KB adalah 8424,25 ms. Berdasarkan grafik yang dihasilkan didapatkan kesimpulan bahwa pada SPECK128/192 semakin besar ukuran *file* yang digunakan maka waktu untuk sistem melakukan proses dekripsi semakin lama. Hal ini disebabkan oleh perbedaan panjang kata dan kalimat yang ada pada masing-masing *file* PDF. Pada *file* PDF dengan ukuran 50 KB menghasilkan panjang kata dan kalimat terpendek dibanding ukuran *file* PDF lainnya. Semakin panjang kata dan kalimat maka semakin panjang proses yang dikerjakan SPECK untuk melakukan dekripsi.

6.2.7 Pengujian Waktu Dekripsi SPECK128/256

Hasil pengujian waktu dekripsi SPECK128/256 dengan varian ukuran *file* ditunjukkan pada Gambar 6.10.

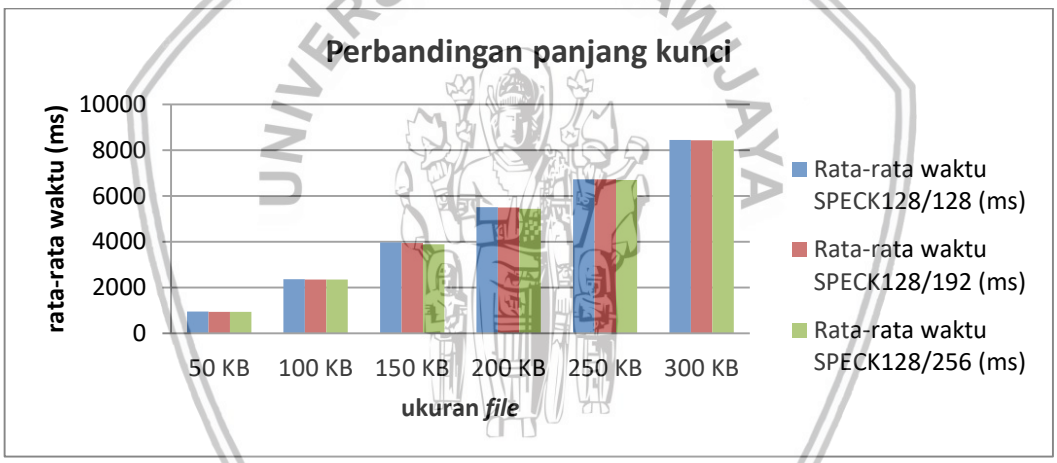


Gambar 6.10 Grafik waktu dekripsi Algoritme SPECK128/256

Pada Gambar 6.10 menunjukkan perbedaan rata-rata waktu dekripsi pada setiap varian ukuran *file*. Pada kiri grafik merupakan keterangan rata-rata waktu dekripsi dan pada bawah grafik merupakan keterangan varian ukuran *file*. Rata-rata waktu yang dihasilkan *file* 50 KB adalah 928,29 ms, *file* 100 KB adalah 2339,29 ms, *file* 150 KB adalah 3876,89 ms, *file* 200 KB adalah 5440,73 ms, *file* 250 KB adalah 6692,63 ms dan *file* 300 KB adalah 8418,45 ms. Berdasarkan grafik yang dihasilkan didapatkan kesimpulan bahwa pada SPECK128/256 semakin besar ukuran *file* yang digunakan maka waktu untuk sistem melakukan proses dekripsi semakin lama. Hal ini disebabkan oleh perbedaan panjang kata dan kalimat yang ada pada masing-masing *file* PDF. Pada *file* PDF dengan ukuran 50 KB menghasilkan panjang kata dan kalimat terpendek dibanding ukuran *file* PDF lainnya. Semakin panjang kata dan kalimat maka semakin panjang proses yang dikerjakan SPECK untuk melakukan dekripsi.

6.2.8 Perbandingan Panjang Kunci SPECK Waktu Dekripsi

Hasil pengujian perbandingan panjang kunci SPECK berdasarkan waktu dekripsi ditunjukkan pada Gambar 6.11.



Gambar 6.11 Grafik perbandingan panjang kunci SPECK waktu dekripsi

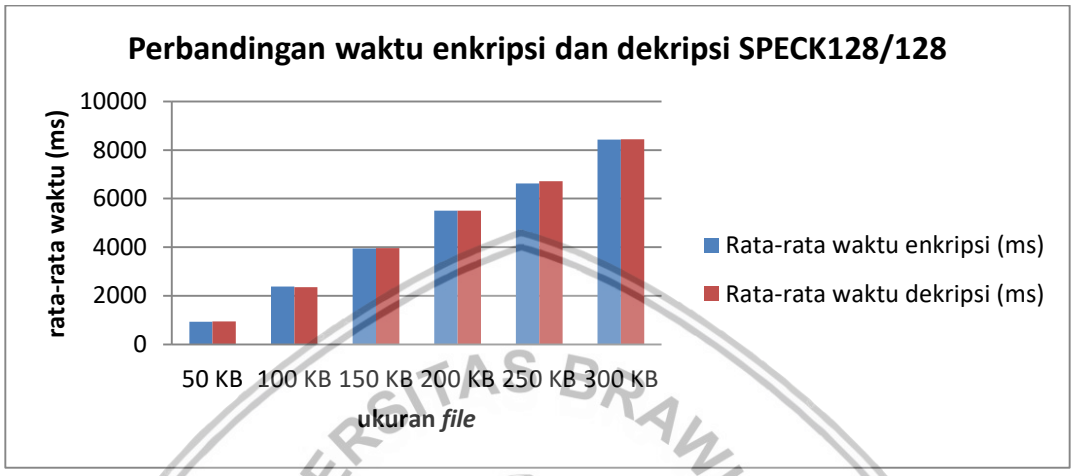
Pada Gambar 6.11 menunjukkan perbandingan rata-rata waktu dekripsi pada setiap varian ukuran *file* berdasarkan panjang kunci yang digunakan. Pada kiri grafik merupakan keterangan rata-rata waktu dekripsi dan pada bawah grafik merupakan keterangan varian ukuran *file*. Berdasarkan grafik yang dihasilkan, SPECK128/256 menghasilkan rata-rata waktu dekripsi tercepat sedangkan SPECK128/128 menghasilkan rata-rata waktu dekripsi paling lama. Kemudian dapat disimpulkan bahwa semakin panjang kunci yang digunakan maka waktu untuk sistem melakukan proses dekripsi semakin cepat.

Hal diatas disebabkan karena sistem membaca masukan dari pengguna berupa karakter di mana semisal 128 bit menghasilkan sejumlah 16 karakter sedangkan 256 bit menghasilkan sejumlah 32 karakter. SPECK yang mengambil ukuran data 256 bit akan menjalankan 32 karakter dalam sekali proses. Hal ini menyebabkan waktu proses pada ukuran 256 bit menjadi lebih cepat dibandingkan dengan ukuran 128 bit di mana hal tersebut juga menyebabkan

SPECK128/256 memiliki waktu proses yang lebih cepat dibanding SPECK128/128 dan SPECK128/192.

6.2.9 Perbandingan Waktu Enkripsi dan Dekripsi SPECK128/128

Hasil perbandingan waktu enkripsi dan dekripsi SPECK128/128 berdasarkan besaran ukuran *file* ditunjukkan pada Gambar 6.12.

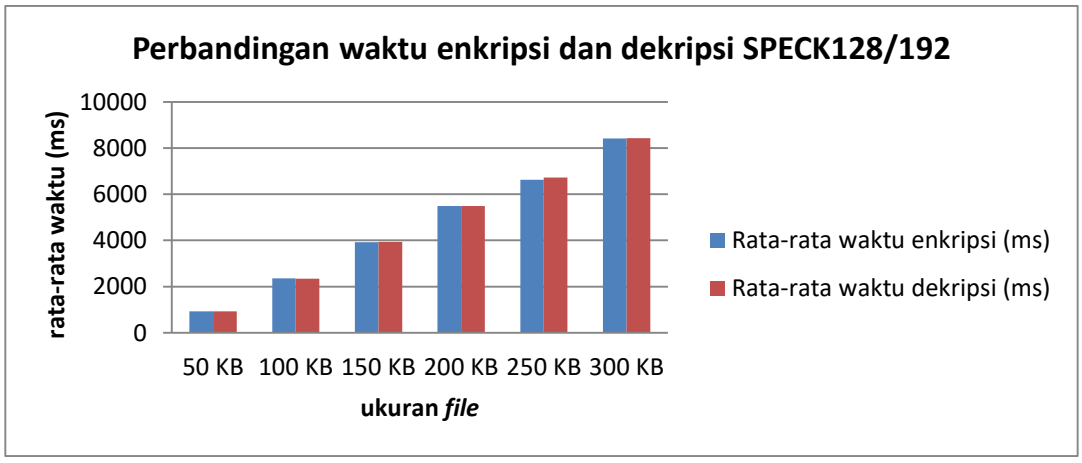


Gambar 6.12 Grafik waktu dekripsi Algoritme SPECK128/128

Pada Gambar 6.12 menunjukkan perbandingan rata-rata waktu enkripsi dan dekripsi pada setiap varian ukuran *file*. Pada kiri grafik merupakan keterangan rata-rata waktu dan pada bawah grafik merupakan keterangan varian ukuran *file*. Berdasarkan grafik yang dihasilkan maka dapat ditarik kesimpulan bahwa antara waktu enkripsi dan dekripsi memiliki perbedaan waktu untuk prosesnya. Perbedaan waktu pada *file* 50 KB sebesar 6,15 ms, *file* 100 KB sebesar 23,53 ms, *file* 150 KB sebesar 14,18 ms, *file* 200 KB sebesar 3,84 ms, *file* 250 KB sebesar 95,45 ms, *file* 300 KB sebesar 5,95 ms.

6.2.10 Perbandingan Waktu Enkripsi dan Dekripsi SPECK128/192

Hasil perbandingan waktu enkripsi dan dekripsi SPECK128/192 berdasarkan besaran ukuran *file* ditunjukkan pada Gambar 6.13.

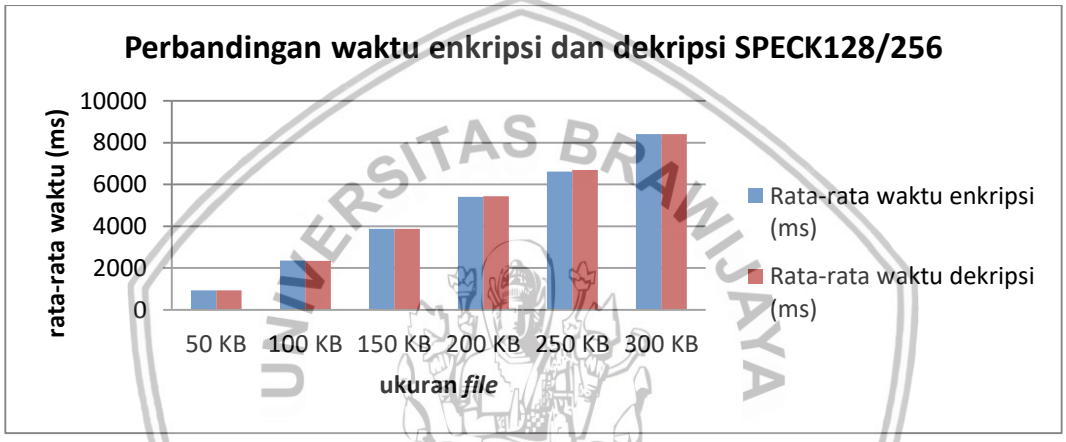


Gambar 6.13 Grafik waktu dekripsi Algoritme SPECK128/192

Pada Gambar 6.13 menunjukkan perbandingan rata-rata waktu enkripsi dan dekripsi pada setiap varian ukuran *file*. Pada kiri grafik merupakan keterangan rata-rata waktu dan pada bawah grafik merupakan keterangan varian ukuran *file*. Berdasarkan grafik yang dihasilkan maka dapat ditarik kesimpulan bahwa antara waktu enkripsi dan dekripsi memiliki perbedaan waktu untuk prosesnya. Perbedaan waktu pada *file* 50 KB sebesar 2,04 ms, *file* 100 KB sebesar 18,001 ms, *file* 150 KB sebesar 16,73 ms, *file* 200 KB sebesar 8,05 ms, *file* 250 KB sebesar 95,15 ms, *file* 300 KB sebesar 2,52 ms.

6.2.11 Perbandingan Waktu Enkripsi dan Dekripsi SPECK128/256

Hasil perbandingan waktu enkripsi dan dekripsi SPECK128/256 berdasarkan besaran ukuran *file* ditunjukkan pada Gambar 6.14.



Gambar 6.14 Grafik waktu dekripsi Algoritme SPECK128/192

Pada Gambar 6.14 menunjukkan perbandingan rata-rata waktu enkripsi dan dekripsi pada setiap varian ukuran *file*. Pada kiri grafik merupakan keterangan rata-rata waktu dan pada bawah grafik merupakan keterangan varian ukuran *file*. Berdasarkan grafik yang dihasilkan maka dapat ditarik kesimpulan bahwa antara waktu enkripsi dan dekripsi memiliki perbedaan waktu untuk prosesnya. Perbedaan waktu pada *file* 50 KB sebesar 1,16 ms, *file* 100 KB sebesar 17,41 ms, *file* 150 KB sebesar 2,37 ms, *file* 200 KB sebesar 34,34 ms, *file* 250 KB sebesar 72,21 ms, *file* 300 KB sebesar 7,38 ms.

6.3 Pengujian *Avalanche Effect*

Pengujian *avalanche effect* bertujuan untuk mengukur tingkat keamanan suatu algoritme kriptografi. Di dalam penelitian akan dimasukkan *plaintext* bernilai 0 dengan *key* bernilai 0 kemudian dibandingkan dengan masukan *plaintext* atau *key* yang mengalami perubahan 1 digit. Penjelasan mengenai hasil analisis pengujian *avalanche effect* akan dijelaskan pada bab Analisis Hasil Pengujian poin 7.3.

6.3.1 Pengujian *Avalanche Effect* SPECK128/128

Perbandingan masukan *plaintext* SPECK128/128 ditunjukkan pada Tabel 6.7.

Tabel 6.7 Perbandingan *plaintext* SPECK128/128

Key	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
Plaintext1	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
Plaintext2	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000001
Ciphertext1	01100110 01011100 00000010 11111101 11011100 11110011 10001101 0111011000100000 10001110 11010111 01001100 00000011 01111111 00001010 01101101
Ciphertext2	01101001 10011011 01110111 11111101 01111010 10100101 00100101 00000100 10101010 00000001 00000001 11101100 10101000 01010101 10110010 11100110

Perbandingan masukan *key* SPECK128/128 ditunjukkan pada Tabel 6.8.

Tabel 6.8 Perbandingan *key* SPECK128/128

Plaintext	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
Key1	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
Key2	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000001
Ciphertext1	01100110 01011100 00000010 11111101 11011100 11110011 10001101 0111011000100000 10001110 11010111 01001100 00000011 01111111 00001010 01101101
Ciphertext2	01010011 10100111 01001110 00110101 11011101 01110011 1100011101010001 10110100 00001001 01001110 01100100 00110001 11001010 10101000 01111101

6.3.2 Pengujian *Avalanche Effect* SPECK128/192

Perbandingan masukan *plaintext* SPECK128/192 ditunjukkan pada Tabel 6.9.

Tabel 6.9 Perbandingan *plaintext* SPECK128/192

Key	00000000 00000000
Plaintext1	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Tabel 6.9 Perbandingan *plaintext* SPECK128/192 (lanjutan)

Plaintext2	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000001
Ciphertext1	10010010 00011101 10011101 01100000 00101100 00100000 10100010 00010011 01000111 10001110 10001111 01111110 11010010 00110011 11110111 00111010
Ciphertext2	01110011 10001110 01110000 00111110 11110000 11110001 11001100 10010010 10010111 11110111 10001110 00000110 01111001 10000101 10000011 10101100

Perbandingan masukan *key* SPECK128/192 ditunjukkan pada Tabel 6.10.

Tabel 6.10 Perbandingan *key* SPECK128/192

Plaintext	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
Key1	00000000 00000000
Key2	00000000 00000001
Ciphertext1	10010010 00011101 10011101 01100000 00101100 00100000 10100010 00010011 01000111 10001110 10001111 01111110 11010010 00110011 11110111 00111010
Ciphertext2	10100110 11011001 00010100 11010011 01010011 10111110 11010000 11111000 10101000 00001000 11111001 10100111 11110111 00101100 10100110 11101111

6.3.3 Pengujian *Avalanche Effect* SPECK128/256

Perbandingan masukan *plaintext* SPECK128/256 ditunjukkan pada Tabel 6.11.

Tabel 6.11 Perbandingan *plaintext* SPECK128/256

Key	00000000 00000000
Plaintext1	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
Plaintext2	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000001

Tabel 6.11 Perbandingan *plaintext* SPECK128/256 (lanjutan)

Ciphertext1	11100010 01001101 10111010 01001110 00111111 10001001 10000101 11011110 00011010 01001011 00111101 10010111 01010101 11011110 10101011 00010110
Ciphertext2	11111110 00100100 01001010 01101001 01001011 11101111 10000101 10111110 10101110 10101011 00101110 01010100 00001100 10111001 00001111 00001101

Perbandingan masukan *key* SPECK128/256 ditunjukkan pada Tabel 6.12.

Tabel 6.12 Perbandingan *key* SPECK128/256

Plaintext	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
Key1	00000000 00000000
Key2	00000000 00000001
Ciphertext1	11100010 01001101 10111010 01001110 00111111 10001001 10000101 11011110 00011010 01001011 00111101 10010111 01010101 11011110 10101011 00010110
Ciphertext2	00100100 10011111 00100110 01111010 11111000 1101010110011101 11011111 00000111 10111101 10000111 11101011 10001001 01101011 10100100 10110010

6.4 Pengujian Waktu Sistem

Pengujian waktu sistem adalah waktu yang dibutuhkan sistem dalam membaca keseluruhan proses saat menjalankan proses algoritme SPECK. Salah satu proses selain adanya waktu enkripsi dan dekripsi adalah proses pembacaan *file* yang diunggah pengguna ke dalam sistem. Hasil pengujian waktu sistem ditampilkan dalam Tabel 6.13.

Tabel 6.13 Hasil pengujian waktu sistem

	50 KB	**100 KB**	**150 KB**	**200 KB**	**250 KB**	**300 KB**
Percobaan-1	91 d	267 d	770 d	789 d	3101 d	6731 d
Percobaan-2	83 d	264 d	781d	791 d	3097 d	6718 d
Percobaan-3	80 d	270 d	776 d	795 d	3102 d	6725 d

Dari hasil pengujian waktu sistem dengan variabel d sebagai detik, waktu proses yang dominan peneliti gunakan adalah waktu yang paling cepat pertama dan kedua yang memakan waktu maksimal 5 menit sehingga peneliti melakukan pengujian waktu enkripsi dan dekripsi sebanyak 1024 kali percobaan hingga pada *file* 100 KB. Pada pengujian 10 menit ke atas bila ditotal dengan varian ketiga SPECK untuk dilakukan pengujian 1024 kali percobaan akan memakan waktu yang begitu lama, maka dari itu untuk varian ukuran *file* lainnya akan dilakukan pengujian waktu enkripsi dan dekripsi sebanyak 170 kali. Hal yang menjadi faktor sistem bekerja untuk keseluruhan proses menghasilkan waktu demikian karena CPU yang digunakan saat sistem meng*upload file* sebesar 24% - 27% dan saat sistem sedang melakukan proses enkripsi dan dekripsi sebesar 52% - 65%. Sedangkan memori RAM yang terpakai sebesar 50% - 52%.

BAB 7 ANALISIS HASIL PENGUJIAN

Bab ini akan menjelaskan analisis dari hasil pengujian yang telah dilakukan yaitu pengujian *test vector* untuk menguji ke*valid*an suatu algoritme, pengujian waktu eksekusi yang menghasilkan nilai kinerja algoritme dan pengujian *avalanche effect* untuk menguji tingkat keamanan.

7.1 Analisis Pengujian *Test Vector*

Pengujian dilakukan dengan memasukkan *input*an berupa *plaintext* dan *key* yang terdapat pada *paper* ke dalam sistem. Hasil dari pengujian *test vector* ditunjukkan pada Tabel 7.1.

Tabel 7.1 Hasil Pengujian *Test Vector*

	***Ciphertext* Hasil Sistem**	***Ciphertext Test Vector***
SPECK128/128	a65d985179783265 7860fedf5c570d18	a65d985179783265 7860fedf5c570d18
SPECK128/192	1be4cf3a13135566 f9bc185de03c1886	1be4cf3a13135566 f9bc185de03c1886
SPECK128/256	4109010405c0f53e 4eeeb48d9c188f43	4109010405c0f53e 4eeeb48d9c188f43

Dari hasil yang ditunjukkan pada Tabel 7.1 menjelaskan bahwa *ciphertext* yang dihasilkan sistem dengan *ciphertext* dari *paper* bernilai sama maka dapat dinyatakan hasil pengujian *test vector* pada sistem adalah *valid*.

7.2 Analisis Pengujian Waktu Eksekusi

Pada bab pengujian telah didapatkan data hasil pengujian yang kemudian data tersebut akan dianalisis untuk menilai kinerja Algoritme SPECK. Hal yang pertama dilakukan adalah analisis perbandingan kunci dengan melakukan pengujian Kolmogorov-Smirnov untuk menilai apakah nilai residual dari data berdistribusi normal atau tidak berdistribusi normal. Apabila nilai residual data tidak berdistribusi normal maka langkah selanjutnya adalah melakukan pengujian Kruskal-Wallis. Pengujian Kruskal-Wallis bertujuan untuk mencari perbedaan waktu antara data yang difokuskan pada data non-parametik. Apabila nilai signifikansi yang dihasilkan dalam pengujian Kruskal-Wallis < 0,05 maka akan dilakukan pengujian *Post Hoc test*. Pengujian *Post Hoc test* bertujuan untuk mencari data kelompok manakah yang menyebabkan terjadinya perbedaan waktu tertinggi dengan melihat nilai signifikansi dan *Chi-Square* tertinggi yang dihasilkan. Data kelompok yang akan dibandingkan adalah data 1 dengan data 2, data 1 dengan data 3 dan data 2 dengan data 3 yang di mana data 1 adalah SPECK128/128, data 2 adalah SPECK128/192 dan data 3 adalah SPECK128/256.

Analisis kedua adalah analisis perbandingan ukuran *file* dengan melakukan pengujian Kolmogorov-Smirnov hingga pengujian *Post hoc test* berdasarkan hipotesis yang telah ditentukan. Data kelompok yang akan dibandingkan adalah 50 KB, 100 KB, 150 KB, 200 KB, 250 KB dan 300 KB.

7.2.1 Analisis *File* 50 KB

Analisis dilakukan dengan memfokuskan pada variasi kunci yang digunakan yaitu 128 bit, 192 bit dan 256 bit untuk mencari variasi kunci mana yang memiliki kinerja terbaik pada *file* 50 KB.

7.2.1.1 Pengujian Kolmogorov-Smirnov *File* 50 KB

Pengujian dilakukan dengan menganalisis nilai residual dari keseluruhan data SPECK128/128, SPECK128/192 dan SPECK128/256 yang berjumlah 1024 data untuk masing-masing waktu enkripsi dan dekripsi. Kemudian akan diambil keputusan berdasarkan nilai signifkansi yang dihasilkan. Hasil pengujian Kolmogorov-Smirnov waktu enkripsi untuk *file* 50 KB ditunjukkan pada Tabel 7.2 dan waktu dekripsi ditunjukkan pada Tabel 7.3.

Tabel 7.2 Hasil pengujian Kolmogorov-Smirnov waktu enkripsi *file* 50 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		3072
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,81555239
Most Extreme Differences	Absolute	,201
	Positive	,194
	Negative	-,201
Test Statistic		,201
Asymp. Sig. (2-tailed)		,000[c]

Tabel 7.3 Hasil pengujian Kolmogorov-Smirnov waktu dekripsi *file* 50 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		3072
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,81446981
Most Extreme Differences	Absolute	,201
	Positive	,186
	Negative	-,201
Test Statistic		,201
Asymp. Sig. (2-tailed)		,000[c]

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka nilai residual berdistribusi normal
H1: nilai signifikansi < 0,05 maka nilai residual tidak berdistribusi normal

Dalam hal ini, nilai signifikansi yang dihasilkan oleh nilai residual waktu enkripsi dan dekripsi < 0,05 maka keputusan H0 ditolak yang artinya nilai residual yang dihasilkan tidak berdistribusi normal. Maka langkah selanjutnya adalah melakukan pengujian Kruskal-Wallis untuk mencari perbedaan waktu antara data.

7.2.1.2 Pengujian Kruskal-Wallis *File* 50 KB

Pengujian dilakukan setelah hasil dari pengujian Kolmogorov-Smirnov memiliki nilai residual tidak berdistribusi normal. Pengujian dilakukan pada data variansi algoritme SPECK yaitu SPECK128/128, SPECK128/192 dan SPECK128/256 berdasarkan waktu enkripsi dan waktu dekripsi yang dimiliki masing-masing sebanyak 1024 data sehingga akan muncul nilai Mean Rank dan nilai signifikansi. Langkah awal dalam pengujian Kruskal-Wallis yaitu menguji varibilitas nilai ujian data variansi algoritme SPECK. Varibilitas yang dimaksud adalah bentuk dan sebaran data. Apabila sebaran data sama maka dapat mengetahui perbedaan *Median* dan *Mean*. Jika tidak memiliki sebaran data yang sama maka hanya dapat mengetahui perbedaan *Mean* saja. Dalam hal ini, hasil yang didapatkan adalah data variansi algoritme SPECK tidak memiliki sebaran data yang sama sehingga hanya dapat mengetahui perbedaan nilai *Mean*. Peringkat rata-rata waktu enkripsi *file* 50 KB ditunjukkan pada Tabel 7.4 dan waktu dekripsi ditunjukkan pada Tabel 7.5.

Tabel 7.4 Rank Kruskal-Wallis waktu enkripsi *file* 50 KB

Ranks

	panjang_kunci	N	Mean Rank
enkripsi	SPECK_128	1024	1596,67
	SPECK_192	1024	1526,40
	SPECK_256	1024	1486,43
	Total	3072	

Tabel 7.5 Rank Kruskal-Wallis waktu dekripsi *file* 50 KB

Ranks

	panjang_kunci	N	Mean Rank
dekripsi	SPECK_128	1024	1613,22
	SPECK_192	1024	1516,52
	SPECK_256	1024	1479,76
	Total	3072	

Langkah selanjutnya adalah analisis Kruskal-Wallis dengan pengambilan keputusan berdasarkan nilai signifikansi yang diperoleh untuk mengukur perbedaan peringkat rata-rata yang didapatkan signifikan atau tidak. Hasil pengujian Kruskal Wallis waktu enkripsi untuk *file* 50 KB akan ditunjukkan pada Tabel 7.6 dan waktu dekripsi ditunjukkan pada Tabel 7.7.

Tabel 7.6 Hasil pengujian Kruskal-Wallis waktu enkripsi *file* 50 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	8,154
df	2
Asymp. Sig.	,017

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Tabel 7.7 Hasil pengujian Kruskal-Wallis waktu dekripsi *file* 50 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	12,416
df	2
Asymp. Sig.	,002

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka tidak terdapat perbedaan waktu antar data kelompok

H1: nilai signifikansi < 0,05 maka terdapat perbedaan waktu antar data kelompok

Hasil yang didapatkan adalah nilai signifikansi < 0,05 maka keputusan H0 ditolak yang berarti terdapat perbedaan waktu antara data variansi algoritme SPECK yang dianalisis. Langkah berikutnya adalah melakukan pengujian *Post Hoc test* untuk mengetahui di mana data kelompok yang menimbulkan perbedaan waktu tertinggi.

7.2.1.3 Pengujian *Post Hoc test File* 50 KB

Pengujian *Post Hoc test* dilakukan pada data kelompok SPECK128/128 dengan SPECK128/192, SPECK128/128 dengan SPECK128/256 dan SPECK128/192 dengan SPECK128/256. Pada Tabel 7.6 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi diantara data

kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu enkripsi *file* 50 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 3,233. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.8.

Tabel 7.8 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/192 *file* 50 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	3,233
df	1
Asymp. Sig.	,072

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 7,951. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.9.

Tabel 7.9 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/256 *file* 50 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	7,951
df	1
Asymp. Sig.	,005

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 1,047. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.10.

Tabel 7.10 Nilai *Chi-Square* waktu enkripsi SPECK128/192 dengan SPECK128/256 *file* 50 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	1,047
df	1
Asymp. Sig.	,306

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 7,951. Nilai signifikansi yang dihasilkan pada Tabel 7.9 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan. Hal ini terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

Selanjutnya pada Tabel 7.7 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu dekripsi *file* 50 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 5,996. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.11.

Tabel 7.11 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/192 *file* 50 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	5,996
df	1
Asymp. Sig.	,014

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 11,785. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.12.

Tabel 7.12 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/256 *file* 50 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	11,785
df	1
Asymp. Sig.	,001

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 0,841. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.13.

Tabel 7.13 Nilai *Chi-Square* waktu dekripsi SPECK128/192 dengan SPECK128/256 *file* 50 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	,841
df	1
Asymp. Sig.	,359

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 11,785. Nilai signifikansi yang dihasilkan pada Tabel 7.12 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan. Hal ini terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

7.2.2 Analisis *File* 100 KB

7.2.2.1 Pengujian Kolmogorov-Smirnov *File* 100 KB

Pengujian dilakukan dengan menganalisis nilai residual dari keseluruhan data SPECK128/128, SPECK128/192 dan SPECK128/256 yang berjumlah 1024 data untuk masing-masing waktu enkripsi dan dekripsi. Kemudian akan diambil keputusan berdasarkan nilai signifkansi yang dihasilkan. Hasil pengujian Kolmogorov-Smirnov waktu enkripsi untuk *file* 100 KB ditunjukkan pada Tabel 7.14 dan waktu dekripsi ditunjukkan pada Tabel 7.15.

Tabel 7.14 Hasil pengujian Kolmogorov-Smirnov waktu enkripsi *file* 100 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		3072
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,81182811
Most Extreme Differences	Absolute	,186
	Positive	,162
	Negative	-,186
Test Statistic		,186
Asymp. Sig. (2-tailed)		,000[c]

Tabel 7.15 Hasil pengujian Kolmogorov-Smirnov waktu dekripsi *file* 100 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		3072
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,81438994
Most Extreme Differences	Absolute	,201
	Positive	,178
	Negative	-,201
Test Statistic		,201
Asymp. Sig. (2-tailed)		,000[c]

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka nilai residual berdistribusi normal
H1: nilai signifikansi < 0,05 maka nilai residual tidak berdistribusi normal

Dalam hal ini, nilai signifikansi yang dihasilkan oleh nilai residual waktu enkripsi dan dekripsi < 0,05 maka keputusan H0 ditolak yang artinya nilai residual yang dihasilkan tidak berdistribusi normal. Maka langkah selanjutnya adalah melakukan pengujian Kruskal-Wallis untuk mencari perbedaan waktu antar data kelompok.

7.2.2.2 Pengujian Kruskal-Wallis *File* 100 KB

Pengujian dilakukan setelah hasil dari pengujian Kolmogorov-Smirnov memiliki nilai residual tidak berdistribusi normal. Pengujian dilakukan pada data variansi algoritme SPECK yaitu SPECK128/128, SPECK128/192 dan SPECK128/256 berdasarkan waktu enkripsi dan waktu dekripsi yang dimiliki masing-masing sebanyak 1024 data sehingga akan muncul nilai Mean Rank dan nilai signifikansi. Langkah awal dalam pengujian Kruskal-Wallis yaitu menguji varibilitas nilai ujian data variansi algoritme SPECK. Varibilitas yang dimaksud adalah bentuk dan sebaran data. Apabila sebaran data sama maka dapat mengetahui perbedaan *Median* dan *Mean*. Jika tidak memiliki sebaran data yang sama maka hanya dapat mengetahui perbedaan *Mean* saja. Dalam hal ini, hasil yang didapatkan adalah data variansi algoritme SPECK tidak memiliki sebaran data yang sama sehingga hanya dapat mengetahui perbedaan nilai *Mean*. Peringkat rata-rata waktu enkripsi *file* 100 KB ditunjukkan pada Tabel 7.16 dan waktu dekripsi ditunjukkan pada Tabel 7.17.

Tabel 7.16 Rank Kruskal-Wallis waktu enkripsi *file* 100 KB

Ranks

	panjang_kunci	N	Mean Rank
Enkripsi	SPECK_128	1024	1637,02
	SPECK_192	1024	1513,50
	SPECK_256	1024	1458,98
	Total	3072	

Tabel 7.17 Rank Kruskal-Wallis waktu dekripsi *file* 100 KB

Ranks

	panjang_kunci	N	Mean Rank
Dekripsi	SPECK_128	1024	1611,93
	SPECK_192	1024	1532,14
	SPECK_256	1024	1465,43
	Total	3072	

Langkah selanjutnya adalah analisis Kruskal-Wallis dengan pengambilan keputusan berdasarkan nilai signifikansi yang diperoleh untuk mengukur perbedaan peringkat rata-rata yang didapatkan signifikan atau tidak. Hasil pengujian Kruskal Wallis waktu enkripsi untuk *file* 100 KB akan ditunjukkan pada Tabel 7.18 dan waktu dekripsi ditunjukkan pada Tabel 7.19.

Tabel 7.18 Hasil pengujian Kruskal-Wallis waktu enkripsi *file* 100 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	21,725
df	2
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Tabel 7.19 Hasil pengujian Kruskal-Wallis waktu dekripsi *file* 100 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	14,069
df	2
Asymp. Sig.	,001

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka tidak terdapat perbedaan waktu antar data kelompok
H1: nilai signifikansi < 0,05 maka terdapat perbedaan waktu antar data kelompok

Hasil yang didapatkan adalah nilai signifikansi < 0,05 maka keputusan H0 ditolak yang berarti terdapat perbedaan waktu antara data variansi algoritme SPECK yang dianalisis. Langkah berikutnya adalah melakukan pengujian *Post Hoc test* untuk mengetahui dimana data kelompok yang menimbulkan perbedaan waktu tertinggi.

7.2.2.3 Pengujian *Post Hoc test File* 100 KB

Pengujian *Post Hoc test* dilakukan pada data kelompok SPECK128/128 dengan SPECK128/192, SPECK128/128 dengan SPECK128/256 dan SPECK128/192 dengan SPECK128/256. Pada Tabel 7.18 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu enkripsi *file* 100 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 10,120. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.20.

Tabel 7.20 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/192 *file* 100 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	10,120
df	1
Asymp. Sig.	,001

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 20,447. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.21.

Tabel 7.21 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/256 *file* 100 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	20,447
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 2,014. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.22.

Tabel 7.22 Nilai *Chi-Square* waktu enkripsi SPECK128/192 dengan SPECK128/256 *file* 100 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	2,014
df	1
Asymp. Sig.	,156

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 20,447. Nilai signifikansi yang dihasilkan pada Tabel 7.21 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan. Hal ini terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

Selanjutnya pada Tabel 7.19 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu dekripsi *file* 100 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 4,162. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.23.

Tabel 7.23 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/192 *file* 100 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	4,162
df	1
Asymp. Sig.	,041

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 14,027. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.24.

Tabel 7.24 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/256 *file* 100 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	14,027
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 2,911. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.25.

Tabel 7.25 Nilai *Chi-Square* waktu dekripsi SPECK128/192 dengan SPECK128/256 *file* 100 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	2,911
df	1
Asymp. Sig.	,088

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 14,027. Nilai signifikansi yang dihasilkan pada Tabel 7.24 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan. Hal ini terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

7.2.3 Analisis *File* 150 KB

7.2.3.1 Pengujian Kolmogorov-Smirnov *File* 150 KB

Pengujian dilakukan dengan menganalisis nilai residual dari keseluruhan data SPECK128/128, SPECK128/192 dan SPECK128/256 yang berjumlah 170 data untuk masing-masing waktu enkripsi dan dekripsi. Kemudian akan diambil keputusan berdasarkan nilai signifkansi yang dihasilkan. Hasil pengujian Kolmogorov-Smirnov waktu enkripsi untuk *file* 150 KB ditunjukkan pada Tabel 7.26 dan waktu dekripsi ditunjukkan pada Tabel 7.27.

Tabel 7.26 Hasil pengujian Kolmogorov-Smirnov waktu enkripsi *file* 150 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		510
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,73922593
Most Extreme Differences	Absolute	,109
	Positive	,083
	Negative	-,109
Test Statistic		,109
Asymp. Sig. (2-tailed)		,000[c]

Tabel 7.27 Hasil pengujian Kolmogorov-Smirnov waktu dekripsi *file* 150 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		510
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,68150872
Most Extreme Differences	Absolute	,110
	Positive	,076
	Negative	-,110
Test Statistic		,110
Asymp. Sig. (2-tailed)		,000[c]

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka nilai residual berdistribusi normal
H1: nilai signifikansi < 0,05 maka nilai residual tidak berdistribusi normal

Dalam hal ini, nilai signifikansi yang dihasilkan oleh nilai residual waktu enkripsi dan dekripsi < 0,05 maka keputusan H0 ditolak yang artinya nilai residual yang dihasilkan tidak berdistribusi normal. Maka langkah selanjutnya adalah melakukan pengujian Kruskal-Wallis untuk mencari perbedaan waktu antar data kelompok.

7.2.3.2 Pengujian Kruskal-Wallis *File* 150 KB

Pengujian dilakukan setelah hasil dari pengujian Kolmogorov-Smirnov memiliki nilai residual tidak berdistribusi normal. Pengujian dilakukan pada data variansi algoritme SPECK yaitu SPECK128/128, SPECK128/192 dan SPECK128/256 berdasarkan waktu enkripsi dan waktu dekripsi yang dimiliki masing-masing sebanyak 170 data sehingga akan muncul nilai Mean Rank dan nilai signifikansi.

Langkah awal dalam pengujian Kruskal-Wallis yaitu menguji varibilitas nilai ujian data variansi algoritme SPECK. Varibilitas yang dimaksud adalah bentuk dan sebaran data. Apabila sebaran data sama maka dapat mengetahui perbedaan *Median* dan *Mean*. Jika tidak memiliki sebaran data yang sama maka hanya dapat mengetahui perbedaan *Mean* saja. Dalam hal ini, hasil yang didapatkan adalah data variansi algoritme SPECK tidak memiliki sebaran data yang sama sehingga hanya dapat mengetahui perbedaan nilai *Mean*. Peringkat rata-rata waktu enkripsi *file* 150 KB ditunjukkan pada Tabel 7.28 dan waktu dekripsi ditunjukkan pada Tabel 7.29.

Tabel 7.28 Rank Kruskal-Wallis waktu enkripsi *file* 150 KB

Ranks

	panjang_kunci	N	Mean Rank
enkripsi	SPECK_128	170	331,23
	SPECK_192	170	260,87
	SPECK_256	170	174,39
	Total	510	

Tabel 7.29 Rank Kruskal-Wallis waktu dekripsi *file* 150 KB

Ranks

	panjang_kunci	N	Mean Rank
dekripsi	SPECK_128	170	347,30
	SPECK_192	170	278,13
	SPECK_256	170	141,07
	Total	510	

Langkah selanjutnya adalah analisis Kruskal-Wallis dengan pengambilan keputusan berdasarkan nilai signifikansi yang diperoleh untuk mengukur perbedaan peringkat rata-rata yang didapatkan signifikan atau tidak. Hasil pengujian Kruskal Wallis waktu enkripsi untuk *file* 150 KB akan ditunjukkan pada Tabel 7.30 dan waktu dekripsi ditunjukkan pada Tabel 7.31.

Tabel 7.30 Hasil pengujian Kruskal-Wallis waktu enkripsi *file* 150 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	96,897
df	2
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Tabel 7.31 Hasil pengujian Kruskal-Wallis waktu dekripsi *file* 150 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	172,896
df	2
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka tidak terdapat perbedaan waktu antara data kelompok
H1: nilai signifikansi < 0,05 maka terdapat perbedaan waktu antara data kelompok

Hasil yang didapatkan adalah nilai signifikansi < 0,05 maka keputusan H0 ditolak yang berarti terdapat perbedaan waktu antara data variansi algoritme SPECK yang dianalisis. Langkah berikutnya adalah melakukan pengujian *Post Hoc test* untuk mengetahui dimana data kelompok yang menimbulkan perbedaan waktu tertinggi.

7.2.3.3 Pengujian *Post Hoc test File* 150 KB

Pengujian *Post Hoc test* dilakukan pada data kelompok SPECK128/128 dengan SPECK128/192, SPECK128/128 dengan SPECK128/256 dan SPECK128/192 dengan SPECK128/256. Pada Tabel 7.30 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu enkripsi *file* 150 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 17,815. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.32.

Tabel 7.32 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/192 *file* 150 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	17,815
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 100,242. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.33.

Tabel 7.33 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/256 *file* 150 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	100,242
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 27,402. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.34.

Tabel 7.34 Nilai *Chi-Square* waktu enkripsi SPECK128/192 dengan SPECK128/256 *file* 150 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	27,402
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 100,242. Nilai signifikansi yang dihasilkan pada Tabel 7.33 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan. Hal ini terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

Selanjutnya pada Tabel 7.31 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu dekripsi *file* 150 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 32,024. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.35.

Tabel 7.35 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/192 *file* 150 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	32,024
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 134,358. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.36.

Tabel 7.36 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/256 *file* 150 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	134,358
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 98,024. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.37.

Tabel 7.37 Nilai *Chi-Square* waktu dekripsi SPECK128/192 dengan SPECK128/256 *file* 150 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	98,024
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 134,358. Nilai signifikansi yang dihasilkan pada Tabel 7.36 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan.

Hal ini terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

7.2.4 Analisis *File* 200 KB

7.2.4.1 Pengujian Kolmogorov-Smirnov *File* 200 KB

Pengujian dilakukan dengan menganalisis nilai residual dari keseluruhan data SPECK128/128, SPECK128/192 dan SPECK128/256 yang berjumlah 170 data untuk masing-masing waktu enkripsi dan dekripsi. Kemudian akan diambil keputusan berdasarkan nilai signifkansi yang dihasilkan. Hasil pengujian Kolmogorov-Smirnov waktu enkripsi untuk *file* 200 KB ditunjukkan pada Tabel 7.38 dan waktu dekripsi ditunjukkan pada Tabel 7.39.

Tabel 7.38 Hasil pengujian Kolmogorov-Smirnov waktu enkripsi *file* 200 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		510
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,66131575
Most Extreme Differences	Absolute	,065
	Positive	,065
	Negative	-,040
Test Statistic		,065
Asymp. Sig. (2-tailed)		,000[c]

Tabel 7.39 Hasil pengujian Kolmogorov-Smirnov waktu dekripsi *file* 200 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		510
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,73089965
Most Extreme Differences	Absolute	,084
	Positive	,084
	Negative	-,074
Test Statistic		,084
Asymp. Sig. (2-tailed)		,000[c]

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka nilai residual berdistribusi normal
H1: nilai signifikansi < 0,05 maka nilai residual tidak berdistribusi normal

Dalam hal ini, nilai signifikansi yang dihasilkan oleh nilai residual waktu enkripsi dan dekripsi < 0,05 maka keputusan H0 ditolak yang artinya nilai residual yang dihasilkan tidak berdistribusi normal. Maka langkah selanjutnya adalah melakukan pengujian Kruskal-Wallis untuk mencari perbedaan waktu antar data kelompok.

7.2.4.2 Pengujian Kruskal-Wallis *File* 200 KB

Pengujian dilakukan setelah hasil dari pengujian Kolmogorov-Smirnov memiliki nilai residual tidak berdistribusi normal. Pengujian dilakukan pada data variansi algoritme SPECK yaitu SPECK128/128, SPECK128/192 dan SPECK128/256 berdasarkan waktu enkripsi dan waktu dekripsi yang dimiliki masing-masing sebanyak 170 data sehingga akan muncul nilai Mean Rank dan nilai signifikansi. Langkah awal dalam pengujian Kruskal-Wallis yaitu menguji varibilitas nilai ujian data variansi algoritme SPECK. Varibilitas yang dimaksud adalah bentuk dan sebaran data. Apabila sebaran data sama maka dapat mengetahui perbedaan *Median* dan *Mean*. Jika tidak memiliki sebaran data yang sama maka hanya dapat mengetahui perbedaan *Mean* saja. Dalam hal ini, hasil yang didapatkan adalah data variansi algoritme SPECK tidak memiliki sebaran data yang sama sehingga hanya dapat mengetahui perbedaan nilai *Mean*. Peringkat rata-rata waktu enkripsi *file* 200 KB ditunjukkan pada Tabel 7.40 dan waktu dekripsi ditunjukkan pada Tabel 7.41.

Tabel 7.40 Rank Kruskal-Wallis waktu enkripsi *file* 200 KB

Ranks

	panjang_kunci	N	Mean Rank
enkripsi	SPECK_128	170	344,86
	SPECK_192	170	289,64
	SPECK_256	170	132,00
	Total	510	

Tabel 7.41 Rank Kruskal-Wallis waktu dekripsi *file* 200 KB

Ranks

	panjang_kunci	N	Mean Rank
dekripsi	SPECK_128	170	326,79
	SPECK_192	170	274,22
	SPECK_256	170	165,49
	Total	510	

Langkah selanjutnya adalah analisis Kruskal-Wallis dengan pengambilan keputusan berdasarkan nilai signifikansi yang diperoleh untuk mengukur perbedaan peringkat rata-rata yang didapatkan signifikan atau tidak. Hasil pengujian Kruskal Wallis waktu enkripsi untuk *file* 200 KB akan ditunjukkan pada Tabel 7.42 dan waktu dekripsi ditunjukkan pada Tabel 7.43.

Tabel 7.42 Hasil pengujian Kruskal-Wallis waktu enkripsi *file* 200 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	191,242
df	2
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Tabel 7.43 Hasil pengujian Kruskal-Wallis waktu dekripsi *file* 200 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	106,172
df	2
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka tidak terdapat perbedaan waktu antara data kelompok

H1: nilai signifikansi < 0,05 maka terdapat perbedaan waktu antara data kelompok

Hasil yang didapatkan adalah nilai signifikansi < 0,05 maka keputusan H0 ditolak yang berarti terdapat perbedaan waktu antara data variansi algoritme SPECK yang dianalisis. Langkah berikutnya adalah melakukan pengujian *Post Hoc test* untuk mengetahui dimana data kelompok yang menimbulkan perbedaan waktu tertinggi.

7.2.4.3 Pengujian *Post Hoc test File* 200 KB

Pengujian *Post Hoc test* dilakukan pada data kelompok SPECK128/128 dengan SPECK128/192, SPECK128/128 dengan SPECK128/256 dan SPECK128/192 dengan SPECK128/256. Pada Tabel 7.42 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi

diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu enkripsi *file* 200 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 15,771. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.44.

Tabel 7.44 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/192 *file* 200 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	15,771
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 163,976. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.45.

Tabel 7.45 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/256 *file* 200 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	163,976
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 107,677. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.46.

Tabel 7.46 Nilai *Chi-Square* waktu enkripsi SPECK128/192 dengan SPECK128/256 *file* 200 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	107,677
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 163,976. Nilai signifikansi yang dihasilkan pada Tabel 7.45 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan. Hal ini terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbeedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

Selanjutnya pada Tabel 7.43 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu dekripsi *file* 200 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 9,887. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.47.

Tabel 7.47 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/192 *file* 200 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	9,887
df	1
Asymp. Sig.	,002

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 105,003. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.48.

Tabel 7.48 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/256 *file* 200 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	105,003
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 44,358. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.49.

Tabel 7.49 Nilai *Chi-Square* waktu dekripsi SPECK128/192 dengan SPECK128/256 *file* 200 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	44,358
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 105,003. Nilai signifikansi yang dihasilkan pada Tabel 7.48 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan. Hal ini terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

7.2.5 Analisis *File* 250 KB

7.2.5.1 Pengujian Kolmogorov-Smirnov *File* 250 KB

Pengujian dilakukan dengan menganalisis nilai residual dari keseluruhan data SPECK128/128, SPECK128/192 dan SPECK128/256 yang berjumlah 170 data untuk masing-masing waktu enkripsi dan dekripsi. Kemudian akan diambil keputusan berdasarkan nilai signifkansi yang dihasilkan. Hasil pengujian Kolmogorov-Smirnov waktu enkripsi untuk *file* 250 KB ditunjukkan pada Tabel 7.50 dan waktu dekripsi ditunjukkan pada Tabel 7.51.

Tabel 7.50 Hasil pengujian Kolmogorov-Smirnov waktu enkripsi *file* 250 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		510
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,81635501
Most Extreme Differences	Absolute	,208
	Positive	,202
	Negative	-,208
Test Statistic		,208
Asymp. Sig. (2-tailed)		,000[c]

Tabel 7.51 Hasil pengujian Kolmogorov-Smirnov waktu dekripsi *file* 250 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		510
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,80587468
Most Extreme Differences	Absolute	,143
	Positive	,143
	Negative	-,126
Test Statistic		,143
Asymp. Sig. (2-tailed)		,000[c]

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka nilai residual berdistribusi normal
H1: nilai signifikansi < 0,05 maka nilai residual tidak berdistribusi normal

Dalam hal ini, nilai signifikansi yang dihasilkan oleh nilai residual waktu enkripsi dan dekripsi < 0,05 maka keputusan H0 ditolak yang artinya nilai residual yang dihasilkan tidak berdistribusi normal. Maka langkah selanjutnya adalah melakukan pengujian Kruskal-Wallis untuk mencari perbedaan waktu antar data kelompok.

7.2.5.2 Pengujian Kruskal-Wallis *File* 250 KB

Pengujian dilakukan setelah hasil dari pengujian Kolmogorov-Smirnov memiliki nilai residual tidak berdistribusi normal. Pengujian dilakukan pada data variansi algoritme SPECK yaitu SPECK128/128, SPECK128/192 dan SPECK128/256 berdasarkan waktu enkripsi dan waktu dekripsi yang dimiliki masing-masing sebanyak 170 data sehingga akan muncul nilai Mean Rank dan nilai signifikansi. Langkah awal dalam pengujian Kruskal-Wallis yaitu menguji varibilitas nilai ujian data variansi algoritme SPECK. Varibilitas yang dimaksud adalah bentuk dan sebaran data. Apabila sebaran data sama maka dapat mengetahui perbedaan *Median* dan *Mean*. Jika tidak memiliki sebaran data yang sama maka hanya dapat mengetahui perbedaan *Mean* saja. Dalam hal ini, hasil yang didapatkan adalah data variansi algoritme SPECK tidak memiliki sebaran data yang sama sehingga hanya dapat mengetahui perbedaan nilai *Mean*. Peringkat rata-rata waktu enkripsi *file* 250 KB ditunjukkan pada Tabel 7.52 dan waktu dekripsi ditunjukkan pada Tabel 7.53.

Tabel 7.52 Rank Kruskal-Wallis waktu enkripsi *file* 250 KB

Ranks

	panjang_kunci	N	Mean Rank
enkripsi	SPECK_128	170	262,46
	SPECK_192	170	263,29
	SPECK_256	170	240,75
	Total	510	

Tabel 7.53 Rank Kruskal-Wallis waktu dekripsi *file* 250 KB

Ranks

	panjang_kunci	N	Mean Rank
dekripsi	SPECK_128	170	271,58
	SPECK_192	170	268,67
	SPECK_256	170	226,24
	Total	510	

Langkah selanjutnya adalah analisis Kruskal-Wallis dengan pengambilan keputusan berdasarkan nilai signifikansi yang diperoleh untuk mengukur perbedaan peringkat rata-rata yang didapatkan signifikan atau tidak. Hasil pengujian Kruskal Wallis waktu enkripsi untuk *file* 250 KB akan ditunjukkan pada Tabel 7.54 dan waktu dekripsi ditunjukkan pada Tabel 7.55.

Tabel 7.54 Hasil pengujian Kruskal-Wallis waktu enkripsi *file* 250 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	2,562
df	2
Asymp. Sig.	,278

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Tabel 7.55 Hasil pengujian Kruskal-Wallis waktu dekripsi *file* 250 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	10,097
df	2
Asymp. Sig.	,006

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka tidak terdapat perbedaan waktu antara data kelompok
H1: nilai signifikansi < 0,05 maka terdapat perbedaan waktu antara data kelompok

Hasil yang didapatkan untuk waktu enkripsi adalah nilai signifikansi > 0,05 maka tidak terdapat perbedaan waktu antara data kelompok yang diujikan. Hal ini disebabkan karena rata-rata waktu yang dihasilkan oleh masing-masing SPECK yang diujikan hanya terdapat sedikit perbedaan dimana rata-rata waktu proses yang dihasilkan oleh SPECK128/128 sebesar 6628,72 ms, untuk SPECK128/192 sebesar 6625,73 ms dan SPECK128/256 sebesar 6620,41 ms. Sedangkan perbedaan waktu yang dihasilkan untuk data kelompok SPECK128/128 dengan SPECK128/192 sebesar 2.99 ms, untuk data kelompok SPECK128/128 dengan SPECK128/256 sebesar 8.31 ms dan perbedaan untuk SPECK128/192 dengan SPECK128/256 sebesar 5.32 ms. Perbedaan kecil yang dihasilkan menyebabkan terjadinya kesimpulan hasil analisis yang menyatakan tidak terdapatnya perbedaan waktu yang signifikan antara data kelompok.

Sedangkan untuk waktu dekripsi menyatakan bahwa nilai signifikansi < 0,05 maka terdapat perbedaan waktu antara data variansi algoritme SPECK yang dianalisis. Langkah berikutnya adalah melakukan pengujian *Post Hoc test* untuk mengetahui dimana data kelompok yang menimbulkan perbedaan waktu tertinggi.

7.2.5.3 Pengujian *Post Hoc test File* 250 KB

Pengujian *Post Hoc test* dilakukan pada data kelompok SPECK128/128 dengan SPECK128/192, SPECK128/128 dengan SPECK128/256 dan SPECK128/192 dengan SPECK128/256. Pada Tabel 7.54 menghasilkan nilai signifikansi > 0,05 yang berarti tidak terdapat perbedaan waktu antara data kelompok. Hasil pengujian *Post Hoc test* waktu enkripsi *file* 250 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 0,025. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.56.

Tabel 7.56 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/192 *file* 250 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	,025
df	1
Asymp. Sig.	,875

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 2,147. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.57.

Tabel 7.57 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/256 *file* 250 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	2,147
df	1
Asymp. Sig.	,143

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 1,703. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.58.

Tabel 7.58 Nilai *Chi-Square* waktu enkripsi SPECK128/192 dengan SPECK128/256 *file* 250 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	1,703
df	1
Asymp. Sig.	,192

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 2,147. Nilai signifikansi yang dihasilkan pada Tabel 7.57 > 0,05 yang berarti tidak terdapat perbedaan waktu yang signifikan. Hal yang menyebabkan nilai *Chi-square* menjadi paling tinggi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

Selanjutnya pada Tabel 7.55 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu dekripsi *file* 250 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 0,057. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.59.

Tabel 7.59 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/192 *file* 250 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	,057
df	1
Asymp. Sig.	,811

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 7,726. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.60.

Tabel 7.60 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/256 *file* 250 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	7,726
df	1
Asymp. Sig.	,005

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 7,358. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.61.

Tabel 7.61 Nilai *Chi-Square* waktu dekripsi SPECK128/192 dengan SPECK128/256 *file* 250 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	7,358
df	1
Asymp. Sig.	,007

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 7,726. Nilai signifikansi yang dihasilkan pada Tabel 7.60 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan. Hal ini

terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

7.2.6 Analisis *File* 300 KB

7.2.6.1 Pengujian Kolmogorov-Smirnov *File* 300 KB

Pengujian dilakukan dengan menganalisis nilai residual dari keseluruhan data SPECK128/128, SPECK128/192 dan SPECK128/256 yang berjumlah 170 data untuk masing-masing waktu enkripsi dan dekripsi. Kemudian akan diambil keputusan berdasarkan nilai signifkansi yang dihasilkan. Hasil pengujian Kolmogorov-Smirnov waktu enkripsi untuk *file* 300 KB ditunjukkan pada Tabel 7.62 dan waktu dekripsi ditunjukkan pada Tabel 7.63.

Tabel 7.62 Hasil pengujian Kolmogorov-Smirnov waktu enkripsi *file* 300 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		510
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,80764915
Most Extreme Differences	Absolute	,167
	Positive	,151
	Negative	-,167
Test Statistic		,167
Asymp. Sig. (2-tailed)		,000[c]

Tabel 7.63 Hasil pengujian Kolmogorov-Smirnov waktu dekripsi *file* 300 KB

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		510
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,80993898
Most Extreme Differences	Absolute	,175
	Positive	,161
	Negative	-,175
Test Statistic		,175
Asymp. Sig. (2-tailed)		,000[c]

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka nilai residual berdistribusi normal
H1: nilai signifikansi < 0,05 maka nilai residual tidak berdistribusi normal

Dalam hal ini, nilai signifikansi yang dihasilkan oleh nilai residual waktu enkripsi dan dekripsi < 0,05 maka keputusan H0 ditolak yang artinya nilai residual yang dihasilkan tidak berdistribusi normal. Maka langkah selanjutnya adalah melakukan pengujian Kruskal-Wallis untuk mencari perbedaan waktu antar data kelompok.

7.2.6.2 Pengujian Kruskal-Wallis *File* 300 KB

Pengujian dilakukan setelah hasil dari pengujian Kolmogorov-Smirnov memiliki nilai residual tidak berdistribusi normal. Pengujian dilakukan pada data variansi algoritme SPECK yaitu SPECK128/128, SPECK128/192 dan SPECK128/256 berdasarkan waktu enkripsi dan waktu dekripsi yang dimiliki masing-masing sebanyak 170 data sehingga akan muncul nilai Mean Rank dan nilai signifikansi. Langkah awal dalam pengujian Kruskal-Wallis yaitu menguji varibilitas nilai ujian data variansi algoritme SPECK. Varibilitas yang dimaksud adalah bentuk dan sebaran data. Apabila sebaran data sama maka dapat mengetahui perbedaan *Median* dan *Mean*. Jika tidak memiliki sebaran data yang sama maka hanya dapat mengetahui perbedaan *Mean* saja. Dalam hal ini, hasil yang didapatkan adalah data variansi algoritme SPECK tidak memiliki sebaran data yang sama sehingga hanya dapat mengetahui perbedaan nilai *Mean*. Peringkat rata-rata waktu enkripsi *file* 300 KB ditunjukkan pada Tabel 7.64 dan waktu dekripsi ditunjukkan pada Tabel 7.65.

Tabel 7.64 Rank Kruskal-Wallis waktu enkripsi *file* 300 KB

Ranks

	panjang_kunci	N	Mean Rank
enkripsi	SPECK_128	170	284,56
	SPECK_192	170	251,68
	SPECK_256	170	230,26
	Total	510	

Tabel 7.65 Rank Kruskal-Wallis waktu dekripsi *file* 300 KB

Ranks

	panjang_kunci	N	Mean Rank
dekripsi	SPECK_128	170	281,59
	SPECK_192	170	249,22
	SPECK_256	170	235,69
	Total	510	

Langkah selanjutnya adalah analisis Kruskal-Wallis dengan pengambilan keputusan berdasarkan nilai signifikansi yang diperoleh untuk mengukur perbedaan peringkat rata-rata yang didapatkan signifikan atau tidak. Hasil pengujian Kruskal Wallis waktu enkripsi untuk *file* 300 KB akan ditunjukkan pada Tabel 7.66 dan waktu dekripsi ditunjukkan pada Tabel 7.67.

Tabel 7.66 Hasil pengujian Kruskal-Wallis waktu enkripsi *file* 300 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	11,741
df	2
Asymp. Sig.	,003

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Tabel 7.67 Hasil pengujian Kruskal-Wallis waktu dekripsi *file* 300 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	8,723
df	2
Asymp. Sig.	,013

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka tidak terdapat perbedaan waktu antara data kelompok

H1: nilai signifikansi < 0,05 maka terdapat perbedaan waktu antara data kelompok

Hasil yang didapatkan adalah nilai signifikansi < 0,05 maka keputusan H0 ditolak yang berarti terdapat perbedaan waktu antara data variansi algoritme SPECK yang dianalisis. Langkah berikutnya adalah melakukan pengujian *Post Hoc test* untuk mengetahui dimana data kelompok yang menimbulkan perbedaan waktu tertinggi.

7.2.6.3 Pengujian *Post Hoc test File* 300 KB

Pengujian *Post Hoc test* dilakukan pada data kelompok SPECK128/128 dengan SPECK128/192, SPECK128/128 dengan SPECK128/256 dan SPECK128/192 dengan SPECK128/256. Pada Tabel 7.66 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan

dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu enkripsi *file* 300 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 4,080. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.68.

Tabel 7.68 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/192 *file* 300 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	4,080
df	1
Asymp. Sig.	,043

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 11,823. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.69.

Tabel 7.69 Nilai *Chi-Square* waktu enkripsi SPECK128/128 dengan SPECK128/256 *file* 300 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	11,823
df	1
Asymp. Sig.	,001

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 1,698. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.70.

Tabel 7.70 Nilai *Chi-Square* waktu enkripsi SPECK128/192 dengan SPECK128/256 *file* 300 KB

Test Statistics[a,b]

	enkripsi
Chi-Square	1,698
df	1
Asymp. Sig.	,193

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 11,823. Nilai signifikansi yang dihasilkan pada Tabel 7.69 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan. Hal ini terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

Selanjutnya pada Tabel 7.67 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan waktu tertinggi diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu dekripsi *file* 300 KB sebagai berikut:

1. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/192 bernilai 4,203. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.71.

Tabel 7.71 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/192 *file* 300 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	4,203
df	1
Asymp. Sig.	,040

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

2. Nilai *Chi-Square* antara SPECK128/128 dengan SPECK128/256 bernilai 8,116. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.72.

Tabel 7.72 Nilai *Chi-Square* waktu dekripsi SPECK128/128 dengan SPECK128/256 *file* 300 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	8,116
df	1
Asymp. Sig.	,004

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

3. Nilai *Chi-Square* antara SPECK128/192 dengan SPECK128/256 bernilai 0,757. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.73.

Tabel 7.73 Nilai *Chi-Square* waktu dekripsi SPECK128/192 dengan SPECK128/256 *file* 300 KB

Test Statistics[a,b]

	dekripsi
Chi-Square	,757
df	1
Asymp. Sig.	,384

a. Kruskal Wallis Test

b. Grouping Variable: panjang_kunci

Nilai *Chi-Square* tertinggi terdapat pada data kelompok SPECK128/128 dengan SPECK128/256 yaitu bernilai 8,116. Nilai signifikansi yang dihasilkan pada Tabel 7.72 < 0,05 yang berarti terdapat perbedaan waktu yang signifikan. Hal ini terjadi karena perbedaan panjang bit *key* dari data kelompok ini lebih besar dibanding dengan data kelompok lainnya. Dalam proses dari algoritme SPECK ini, semakin panjang perbedaan bit *key* yang digunakan maka waktu yang dibutuhkan untuk memproses algoritme SPECK semakin bertambah.

Dari hasil analisis masing-masing varian ukuran *file* pada algoritme SPECK128/128, SPECK128/192 dan SPECK128/256 dapat ditarik kesimpulan bahwa yang menyebabkan perbedaan waktu tertinggi dimana hal tersebut mempengaruhi nilai signifikansi pada pengujian Kruskal-Wallis adalah data kelompok SPECK128/128 dengan SPECK128/256 karena perbedaan panjang *key* yang digunakan keduanya adalah yang paling besar mencapai dua kali lipat dibanding dengan data kelompok SPECK128/128 dengan SPECK128/192 dan SPECK128/192 dengan SPECK128/256 dimana menyebabkan proses enkripsi dan dekripsi yang terjadi menjadi lebih lama.

7.2.7 Analisis Berdasarkan Ukuran *File*

Analisis kedua dilakukan dengan memfokuskan pada besaran ukuran *file* yang digunakan yaitu 50 KB, 100 KB, 150 KB, 200 KB, 250 KB dan 300 KB untuk mencari kinerja SPECK berdasarkan ukuran *file* yang diujikan.

7.2.7.1 Pengujian Kolmogorov-Smirnov SPECK128/128

Pengujian dilakukan dengan menganalisis nilai residual dari keseluruhan ukuran *file* yang berjumlah 1024 data untuk *file* 50 KB dan 100 KB serta 170 data untuk *file* ukuran lainnya pada masing-masing waktu enkripsi dan dekripsi. Kemudian akan diambil keputusan berdasarkan nilai signifikansi yang dihasilkan. Hasil pengujian Kolmogorov-Smirnov waktu enkripsi pada SPECK128/128 ditunjukkan pada Tabel 7.74 dan waktu dekripsi ditunjukkan pada Tabel 7.75.

Tabel 7.74 Hasil pengujian Kolmogorov-Smirnov waktu enkripsi SPECK128/128

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		2728
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,06522378
Most Extreme Differences	Absolute	,061
	Positive	,061
	Negative	-,036
Test Statistic		,061
Asymp. Sig. (2-tailed)		,000[c]

Tabel 7.75 Hasil pengujian Kolmogorov-Smirnov waktu dekripsi SPECK128/128

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		2728
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,06221216
Most Extreme Differences	Absolute	,059
	Positive	,053
	Negative	-,059
Test Statistic		,059
Asymp. Sig. (2-tailed)		,000[c]

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka nilai residual berdistribusi normal
H1: nilai signifikansi < 0,05 maka nilai residual tidak berdistribusi normal

Dalam hal ini, nilai signifikansi yang dihasilkan oleh nilai residual waktu enkripsi dan dekripsi < 0,05 maka keputusan H0 ditolak yang artinya nilai residual yang dihasilkan tidak berdistribusi normal. Maka langkah selanjutnya adalah melakukan pengujian Kruskal-Wallis untuk mencari perbedaan waktu antara data.

7.2.7.2 Pengujian Kruskal-Wallis SPECK128/128

Pengujian dilakukan setelah hasil dari pengujian Kolmogorov-Smirnov memiliki nilai residual tidak berdistribusi normal. Pengujian dilakukan pada data variansi ukuran *file* yaitu 50 KB, 100 KB, 150 KB, 200 KB, 250 KB dan 300 KB berdasarkan waktu enkripsi dan waktu dekripsi yang dimiliki masing-masing

sebanyak 1024 data untuk ukuran *file* 50 KB dan 100 KB serta 170 data untuk ukuran *file* lainnya sehingga akan muncul nilai Mean Rank dan nilai signifikansi.

Langkah awal dalam pengujian Kruskal-Wallis yaitu menguji varibilitas nilai ujian data variansi ukuran *file*. Varibilitas yang dimaksud adalah bentuk dan sebaran data. Apabila sebaran data sama maka dapat mengetahui perbedaan *Median* dan *Mean*. Jika tidak memiliki sebaran data yang sama maka hanya dapat mengetahui perbedaan *Mean* saja. Dalam hal ini, hasil yang didapatkan adalah data variansi ukuran *file* tidak memiliki sebaran data yang sama sehingga hanya dapat mengetahui perbedaan nilai *Mean*. Peringkat rata-rata waktu enkripsi SPECK128/128 ditunjukkan pada Tabel 7.76 dan waktu dekripsi ditunjukkan pada Tabel 7.77.

Tabel 7.76 Rank Kruskal-Wallis waktu enkripsi SPECK128/128

Ranks

	ukuran_file	N	Mean Rank
enkripsi	50 KB	1024	512,50
	100 KB	1024	1536,50
	150 KB	170	2133,50
	200 KB	170	2303,50
	250 KB	170	2473,50
	300 KB	170	2643,50
	Total	2728	

Tabel 7.77 Rank Kruskal-Wallis waktu dekripsi SPECK128/128

Ranks

	ukuran_file	N	Mean Rank
dekripsi	50 KB	1024	512,50
	100 KB	1024	1536,50
	150 KB	170	2133,50
	200 KB	170	2303,50
	250 KB	170	2473,50
	300 KB	170	2643,50
	Total	2728	

Langkah selanjutnya adalah analisis Kruskal-Wallis dengan pengambilan keputusan berdasarkan nilai signifikansi yang diperoleh untuk mengukur perbedaan peringkat rata-rata yang didapatkan signifikan atau tidak. Hasil pengujian Kruskal Wallis waktu enkripsi untuk SPECK128/128 akan ditunjukkan pada Tabel 7.78 dan waktu dekripsi ditunjukkan pada Tabel 7.79.

Tabel 7.78 Hasil pengujian Kruskal-Wallis waktu enkripsi SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	2437,073
df	5
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Tabel 7.79 Hasil pengujian Kruskal-Wallis waktu dekripsi SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	2436,891
df	5
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka tidak terdapat perbedaan waktu antara data kelompok
H1: nilai signifikansi < 0,05 maka terdapat perbedaan waktu antara data kelompok

Hasil yang didapatkan adalah nilai signifikansi < 0,05 maka keputusan H0 ditolak yang berarti terdapat perbedaan waktu antara data variansi ukuran *file* yang dianalisis. Langkah berikutnya adalah melakukan pengujian *Post Hoc test* untuk mengetahui perbandingan signifikan yang dihasilkan setiap data kelompok.

7.2.7.3 Pengujian *Post Hoc test* SPECK128/128

Pengujian *Post Hoc test* dilakukan pada 15 data kelompok dengan membandingkan masing-masing dua ukuran *file*. Pada Tabel 7.78 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari perbedaan siginifikan diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu enkripsi SPECK128/128 sebagai berikut:

1. Nilai *Chi-Square* antara 50 KB dengan 100 KB bernilai 1536,981. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.80.

Tabel 7.80 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 100 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	1536,981
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

2. Nilai *Chi-Square* antara 50 KB dengan 150 KB bernilai 438,840. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.81.

Tabel 7.81 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 150 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	438,840
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

3. Nilai *Chi-Square* antara 50 KB dengan 200 KB bernilai 438,839. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.82.

Tabel 7.82 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 200 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	438,839
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

4. Nilai *Chi-Square* antara 50 KB dengan 250 KB bernilai 438,838. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.83.

Tabel 7.83 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 250 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	438,838
df	1
Asymp. Sig.	,000

5. Nilai *Chi-Square* antara 50 KB dengan 300 KB bernilai 438,838. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.84.

Tabel 7.84 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 300 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	438,838
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

6. Nilai *Chi-Square* antara 100 KB dengan 150 KB bernilai 437,706. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.85.

Tabel 7.85 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 150 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	437,706
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

7. Nilai *Chi-Square* antara 100 KB dengan 200 KB bernilai 437,705. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.86.

Tabel 7.86 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 200 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	437,705
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

8. Nilai *Chi-Square* antara 100 KB dengan 250 KB bernilai 437,704. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.87.

Tabel 7.87 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 250 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	437,704
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

9. Nilai *Chi-Square* antara 100 KB dengan 300 KB bernilai 437,704. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.88.

Tabel 7.88 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 300 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	437,704
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

10. Nilai *Chi-Square* antara 150 KB dengan 200 KB bernilai 254,531. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.89.

Tabel 7.89 Nilai *Chi-Square* waktu enkripsi 150 KB dengan 200 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	254,531
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

11. Nilai *Chi-Square* antara 150 KB dengan 250 KB bernilai 254,494. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.90.

Tabel 7.90 Nilai *Chi-Square* waktu enkripsi 150 KB dengan 250 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	254,494
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

12. Nilai *Chi-Square* antara 150 KB dengan 300 KB bernilai 254,496. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.91.

Tabel 7.91 Nilai *Chi-Square* waktu enkripsi 150 KB dengan 300 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	254,496
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

13. Nilai *Chi-Square* antara 200 KB dengan 250 KB bernilai 254,463. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.92.

Tabel 7.92 Nilai *Chi-Square* waktu enkripsi 200 KB dengan 250 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	254,463
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

14. Nilai *Chi-Square* antara 200 KB dengan 300 KB bernilai 254,466. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.93.

Tabel 7.93 Nilai *Chi-Square* waktu enkripsi 200 KB dengan 300 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	254,466
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

15. Nilai *Chi-Square* antara 250 KB dengan 300 KB bernilai 254,429. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.94.

Tabel 7.94 Nilai *Chi-Square* waktu enkripsi 250 KB dengan 300 KB SPECK128/128

Test Statistics[a,b]

	enkripsi
Chi-Square	254,429
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Berdasarkan hasil pengujian *Post hoc test*, didapatkan hasil bahwa setiap data kelompok yang diujikan menghasilkan nilai signifikansi < 0,05 semua yang berarti bahwa di setiap data kelompok terdapat perbedaan waktu yang signifikan. Maka dapat disimpulkan bahwa perubahan data setiap 50 KB menghasilkan perubahan waktu yang signifikan. Semisal perubahan 50 KB ke 100 KB dengan selisih ukuran 50 KB dapat menghasilkan perbedaan waktu yang signifikan dan begitu pula untuk ukuran *file* lainnya.

Selanjutnya pada Tabel 7.79 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk untuk mencari perbedaan siginifikan diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu dekripsi SPECK128/128 sebagai berikut:

1. Nilai *Chi-Square* antara 50 KB dengan 100 KB bernilai 1536,705. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.95.

Tabel 7.95 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 100 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	1536,705
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

2. Nilai *Chi-Square* antara 50 KB dengan 150 KB bernilai 438,041. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.96.

Tabel 7.96 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 150 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	438,041
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

3. Nilai *Chi-Square* antara 50 KB dengan 200 KB bernilai 438,036. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.97.

Tabel 7.97 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 200 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	438,036
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

4. Nilai *Chi-Square* antara 50 KB dengan 250 KB bernilai 438,034. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.98.

Tabel 7.98 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 250 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	438,034
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

5. Nilai *Chi-Square* antara 50 KB dengan 300 KB bernilai 438,033. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.99.

Tabel 7.99 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 300 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	438,033
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

6. Nilai *Chi-Square* antara 100 KB dengan 150 KB bernilai 438,117. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.100.

Tabel 7.100 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 150 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	438,117
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

7. Nilai *Chi-Square* antara 100 KB dengan 200 KB bernilai 438,112. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.101.

Tabel 7.101 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 200 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	438,112
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

8. Nilai *Chi-Square* antara 100 KB dengan 250 KB bernilai 438,110. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.102.

Tabel 7.102 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 250 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	438,110
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

9. Nilai *Chi-Square* antara 100 KB dengan 300 KB bernilai 438,109. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.103.

Tabel 7.103 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 300 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	438,109
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

10. Nilai *Chi-Square* antara 150 KB dengan 200 KB bernilai 254,706. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.104.

Tabel 7.104 Nilai *Chi-Square* waktu dekripsi 150 KB dengan 200 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	254,706
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

11. Nilai *Chi-Square* antara 150 KB dengan 250 KB bernilai 254,647. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.105.

Tabel 7.105 Nilai *Chi-Square* waktu dekripsi 150 KB dengan 250 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	254,647
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

12. Nilai *Chi-Square* antara 150 KB dengan 300 KB bernilai 254,612. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.106.

Tabel 7.106 Nilai *Chi-Square* waktu dekripsi 150 KB dengan 300 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	254,612
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

13. Nilai *Chi-Square* antara 200 KB dengan 250 KB bernilai 254,537. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.107.

Tabel 7.107 Nilai *Chi-Square* waktu dekripsi 200 KB dengan 250 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	254,537
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

14. Nilai *Chi-Square* antara 200 KB dengan 300 KB bernilai 254,502. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.108.

Tabel 7.108 Nilai *Chi-Square* waktu dekripsi 200 KB dengan 300 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	254,502
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

15. Nilai *Chi-Square* antara 250 KB dengan 300 KB bernilai 254,443. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.109.

Tabel 7.109 Nilai *Chi-Square* waktu dekripsi 250 KB dengan 300 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	254,443
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Berdasarkan hasil pengujian *Post hoc test*, didapatkan hasil bahwa setiap data kelompok yang diujikan menghasilkan nilai signifikansi < 0,05 semua yang berarti bahwa di setiap data kelompok terdapat perbedaan waktu yang signifikan. Maka dapat disimpulkan bahwa perubahan data setiap 50 KB

menghasilkan perubahan waktu yang signifikan. Semisal perubahan 50 KB ke 100 KB dengan selisih ukuran 50 KB dapat menghasilkan perbedaan waktu yang signifikan dan begitu pula untuk ukuran *file* lainnya.

7.2.7.4 Pengujian Kolmogorov-Smirnov SPECK128/192

Pengujian dilakukan dengan menganalisis nilai residual dari keseluruhan ukuran *file* yang berjumlah 1024 data untuk *file* 50 KB dan 100 KB serta 170 data untuk *file* ukuran lainnya pada masing-masing waktu enkripsi dan dekripsi. Kemudian akan diambil keputusan berdasarkan nilai signifikansi yang dihasilkan. Hasil pengujian Kolmogorov-Smirnov waktu enkripsi pada SPECK128/192 ditunjukkan pada Tabel 7.110 dan waktu dekripsi ditunjukkan pada Tabel 7.111.

Tabel 7.110 Hasil pengujian Kolmogorov-Smirnov waktu enkripsi SPECK128/192

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		2728
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,06237294
Most Extreme Differences	Absolute	,063
	Positive	,063
	Negative	-,040
Test Statistic		,063
Asymp. Sig. (2-tailed)		,000[c]

Tabel 7.111 Hasil pengujian Kolmogorov-Smirnov waktu dekripsi SPECK128/192

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		2728
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,05894021
Most Extreme Differences	Absolute	,067
	Positive	,055
	Negative	-,067
Test Statistic		,067
Asymp. Sig. (2-tailed)		,000[c]

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka nilai residual berdistribusi normal
H1: nilai signifikansi < 0,05 maka nilai residual tidak berdistribusi normal

Dalam hal ini, nilai signifikansi yang dihasilkan oleh nilai residual waktu enkripsi dan dekripsi < 0,05 maka keputusan H0 ditolak yang artinya nilai residual yang dihasilkan tidak berdistribusi normal. Maka langkah selanjutnya adalah melakukan pengujian Kruskal-Wallis untuk mencari perbedaan waktu antara data.

7.2.7.5 Pengujian Kruskal-Wallis SPECK128/192

Pengujian dilakukan setelah hasil dari pengujian Kolmogorov-Smirnov memiliki nilai residual tidak berdistribusi normal. Pengujian dilakukan pada data variansi ukuran *file* yaitu 50 KB, 100 KB, 150 KB, 200 KB, 250 KB dan 300 KB berdasarkan waktu enkripsi dan waktu dekripsi yang dimiliki masing-masing sebanyak 1024 data untuk ukuran *file* 50 KB dan 100 KB serta 170 data untuk ukuran *file* lainnya sehingga akan muncul nilai Mean Rank dan nilai signifikansi.

Langkah awal dalam pengujian Kruskal-Wallis yaitu menguji varibilitas nilai ujian data variansi ukuran *file*. Varibilitas yang dimaksud adalah bentuk dan sebaran data. Dalam hal ini, hasil yang didapatkan adalah data variansi ukuran *file* tidak memiliki sebaran data yang sama sehingga hanya dapat mengetahui perbedaan nilai *Mean*. Peringkat rata-rata waktu enkripsi SPECK128/192 ditunjukkan pada Tabel 7.112 dan waktu dekripsi ditunjukkan pada Tabel 7.113.

Tabel 7.112 Rank Kruskal-Wallis waktu enkripsi SPECK128/192

Ranks

	ukuran_file	N	Mean Rank
enkripsi	50 KB	1024	512,50
	100 KB	1024	1536,50
	150 KB	170	2133,50
	200 KB	170	2303,50
	250 KB	170	2473,50
	300 KB	170	2643,50
	Total	2728	

Tabel 7.113 Rank Kruskal-Wallis waktu dekripsi SPECK128/192

Ranks

	ukuran_file	N	Mean Rank
dekripsi	50 KB	1024	512,50
	100 KB	1024	1536,50
	150 KB	170	2133,50
	200 KB	170	2303,50
	250 KB	170	2473,50
	300 KB	170	2643,50
	Total	2728	

Langkah selanjutnya adalah analisis Kruskal-Wallis dengan pengambilan keputusan berdasarkan nilai signifikansi yang diperoleh untuk mengukur perbedaan peringkat rata-rata yang didapatkan signifikan atau tidak. Hasil pengujian Kruskal Wallis waktu enkripsi untuk SPECK128/192 akan ditunjukkan pada Tabel 7.114 dan waktu dekripsi ditunjukkan pada Tabel 7.115.

Tabel 7.114 Hasil pengujian Kruskal-Wallis waktu enkripsi SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	2436,971
df	5
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Tabel 7.115 Hasil pengujian Kruskal-Wallis waktu dekripsi SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	2436,988
df	5
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka tidak terdapat perbedaan waktu antara data kelompok

H1: nilai signifikansi < 0,05 maka terdapat perbedaan waktu antara data kelompok

Hasil yang didapatkan adalah nilai signifikansi < 0,05 maka keputusan H0 ditolak yang berarti terdapat perbedaan waktu antara data variansi ukuran *file* yang dianalisis. Langkah berikutnya adalah melakukan pengujian *Post Hoc test* untuk mengetahui perbandingan signifikan yang dihasilkan setiap data kelompok.

7.2.7.6 Pengujian *Post Hoc test* SPECK128/192

Pengujian *Post Hoc test* dilakukan pada 15 data kelompok dengan membandingkan masing-masing dua ukuran *file*. Pada Tabel 7.114 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari

perbedaan siginifikan diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu enkripsi SPECK128/192 sebagai berikut:

1. Nilai *Chi-Square* antara 50 KB dengan 100 KB bernilai 1536,826. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.116.

Tabel 7.116 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 100 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	1536,826
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

2. Nilai *Chi-Square* antara 50 KB dengan 150 KB bernilai 438,464. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.117.

Tabel 7.117 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 150 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	438,464
Df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

3. Nilai *Chi-Square* antara 50 KB dengan 200 KB bernilai 438,458. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.118.

Tabel 7.118 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 200 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	438,458
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

4. Nilai *Chi-Square* antara 50 KB dengan 250 KB bernilai 438,463. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.119.

Tabel 7.119 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 250 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	438,463
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

5. Nilai *Chi-Square* antara 50 KB dengan 300 KB bernilai 438,461. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.120.

Tabel 7.120 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 300 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	438,461
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

6. Nilai *Chi-Square* antara 100 KB dengan 150 KB bernilai 437,862. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.121.

Tabel 7.121 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 150 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	437,862
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

7. Nilai *Chi-Square* antara 100 KB dengan 200 KB bernilai 437,856. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.122.

Tabel 7.122 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 200 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	437,856
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

8. Nilai *Chi-Square* antara 100 KB dengan 250 KB bernilai 437,861. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.123.

Tabel 7.123 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 250 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	437,861
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

9. Nilai *Chi-Square* antara 100 KB dengan 300 KB bernilai 437,859. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.124.

Tabel 7.124 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 300 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	437,859
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

10. Nilai *Chi-Square* antara 150 KB dengan 200 KB bernilai 254,526. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.125.

Tabel 7.125 Nilai *Chi-Square* waktu enkripsi 150 KB dengan 200 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	254,526
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

11. Nilai *Chi-Square* antara 150 KB dengan 250 KB bernilai 254,654. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.126.

Tabel 7.126 Nilai *Chi-Square* waktu enkripsi 150 KB dengan 250 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	254,654
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

12. Nilai *Chi-Square* antara 150 KB dengan 300 KB bernilai 254,606. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.127.

Tabel 7.127 Nilai *Chi-Square* waktu enkripsi 150 KB dengan 300 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	254,606
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

13. Nilai *Chi-Square* antara 200 KB dengan 250 KB bernilai 254,497. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.128.

Tabel 7.128 Nilai *Chi-Square* waktu enkripsi 200 KB dengan 250 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	254,497
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

14. Nilai *Chi-Square* antara 200 KB dengan 300 KB bernilai 254,449. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.129.

Tabel 7.129 Nilai *Chi-Square* waktu enkripsi 200 KB dengan 300 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	254,449
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

15. Nilai *Chi-Square* antara 250 KB dengan 300 KB bernilai 254,576. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.130.

Tabel 7.130 Nilai *Chi-Square* waktu enkripsi 250 KB dengan 300 KB SPECK128/192

Test Statistics[a,b]

	enkripsi
Chi-Square	254,576
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Berdasarkan hasil pengujian *Post hoc test*, didapatkan hasil bahwa setiap data kelompok yang diujikan menghasilkan nilai signifikansi < 0,05 semua yang berarti bahwa di setiap data kelompok terdapat perbedaan waktu yang signifikan. Maka dapat disimpulkan bahwa perubahan data setiap 50 KB menghasilkan perubahan waktu yang signifikan. Semisal perubahan 50 KB ke 100 KB dengan selisih ukuran 50 KB dapat menghasilkan perbedaan waktu yang signifikan dan begitu pula untuk ukuran *file* lainnya.

Selanjutnya pada Tabel 7.115 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk untuk mencari perbedaan siginifikan diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu dekripsi SPECK128/192 sebagai berikut:

1. Nilai *Chi-Square* antara 50 KB dengan 100 KB bernilai 1536,854. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.131.

Tabel 7.131 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 100 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	1536,854
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

2. Nilai *Chi-Square* antara 50 KB dengan 150 KB bernilai 438,083. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.132.

Tabel 7.132 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 150 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	438,083
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

3. Nilai *Chi-Square* antara 50 KB dengan 200 KB bernilai 438,080. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.133.

Tabel 7.133 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 200 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	438,080
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

4. Nilai *Chi-Square* antara 50 KB dengan 250 KB bernilai 438,079. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.134.

Tabel 7.134 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 250 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	438,079
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

5. Nilai *Chi-Square* antara 50 KB dengan 300 KB bernilai 438,083. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.135.

Tabel 7.135 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 300 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	438,083
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

6. Nilai *Chi-Square* antara 100 KB dengan 150 KB bernilai 438,280. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.136.

Tabel 7.136 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 150 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	438,280
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

7. Nilai *Chi-Square* antara 100 KB dengan 200 KB bernilai 438,277. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.137.

Tabel 7.137 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 200 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	438,277
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

8. Nilai *Chi-Square* antara 100 KB dengan 250 KB bernilai 438,276. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.138.

Tabel 7.138 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 250 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	438,276
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

9. Nilai *Chi-Square* antara 100 KB dengan 300 KB bernilai 438,279. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.139.

Tabel 7.139 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 300 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	438,279
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

10. Nilai *Chi-Square* antara 150 KB dengan 200 KB bernilai 254,526. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.140.

Tabel 7.140 Nilai *Chi-Square* waktu dekripsi 150 KB dengan 200 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	254,526
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

11. Nilai *Chi-Square* antara 150 KB dengan 250 KB bernilai 254,498. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.141.

Tabel 7.141 Nilai *Chi-Square* waktu dekripsi 150 KB dengan 250 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	254,498
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

12. Nilai *Chi-Square* antara 150 KB dengan 300 KB bernilai 254,596. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.142.

Tabel 7.142 Nilai *Chi-Square* waktu dekripsi 150 KB dengan 300 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	254,596
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

13. Nilai *Chi-Square* antara 200 KB dengan 250 KB bernilai 254,415. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.143.

Tabel 7.143 Nilai *Chi-Square* waktu dekripsi 200 KB dengan 250 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	254,415
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

14. Nilai *Chi-Square* antara 200 KB dengan 300 KB bernilai 254,514. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.144.

Tabel 7.144 Nilai *Chi-Square* waktu dekripsi 200 KB dengan 300 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	254,514
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

15. Nilai *Chi-Square* antara 250 KB dengan 300 KB bernilai 254,486. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.145.

Tabel 7.145 Nilai *Chi-Square* waktu dekripsi 250 KB dengan 300 KB SPECK128/192

Test Statistics[a,b]

	dekripsi
Chi-Square	254,486
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Berdasarkan hasil pengujian *Post hoc test*, didapatkan hasil bahwa setiap data kelompok yang diujikan menghasilkan nilai signifikansi < 0,05 semua yang berarti bahwa di setiap data kelompok terdapat perbedaan waktu yang signifikan. Maka dapat disimpulkan bahwa perubahan data setiap 50 KB menghasilkan perubahan waktu yang signifikan. Semisal perubahan 50 KB ke 100

KB dengan selisih ukuran 50 KB dapat menghasilkan perbedaan waktu yang signifikan dan begitu pula untuk ukuran *file* lainnya.

7.2.7.7 Pengujian Kolmogorov-Smirnov SPECK128/256

Pengujian dilakukan dengan menganalisis nilai residual dari keseluruhan ukuran *file* yang berjumlah 1024 data untuk *file* 50 KB dan 100 KB serta 170 data untuk *file* ukuran lainnya pada masing-masing waktu enkripsi dan dekripsi. Kemudian akan diambil keputusan berdasarkan nilai signifikansi yang dihasilkan. Hasil pengujian Kolmogorov-Smirnov waktu enkripsi pada SPECK128/256 ditunjukkan pada Tabel 7.146 dan waktu dekripsi ditunjukkan pada Tabel 7.147.

Tabel 7.146 Hasil pengujian Kolmogorov-Smirnov waktu enkripsi SPECK128/256

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		2728
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,05736622
Most Extreme Differences	Absolute	,073
	Positive	,073
	Negative	-,048
Test Statistic		,073
Asymp. Sig. (2-tailed)		,000[c]

Tabel 7.147 Hasil pengujian Kolmogorov-Smirnov waktu dekripsi SPECK128/256

One-Sample Kolmogorov-Smirnov Test

		Unstandardized Residual
N		2728
Normal Parameters[a,b]	Mean	,0000000
	Std. Deviation	,05800940
Most Extreme Differences	Absolute	,064
	Positive	,051
	Negative	-,064
Test Statistic		,064
Asymp. Sig. (2-tailed)		,000[c]

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka nilai residual berdistribusi normal
H1: nilai signifikansi < 0,05 maka nilai residual tidak berdistribusi normal

Dalam hal ini, nilai signifikansi yang dihasilkan oleh nilai residual waktu enkripsi dan dekripsi < 0,05 maka keputusan H0 ditolak yang artinya nilai residual yang dihasilkan tidak berdistribusi normal. Maka langkah selanjutnya adalah melakukan pengujian Kruskal-Wallis untuk mencari perbedaan waktu antara data.

7.2.7.8 Pengujian Kruskal-Wallis SPECK128/256

Pengujian dilakukan setelah hasil dari pengujian Kolmogorov-Smirnov memiliki nilai residual tidak berdistribusi normal. Pengujian dilakukan pada data variansi ukuran *file* yaitu 50 KB, 100 KB, 150 KB, 200 KB, 250 KB dan 300 KB berdasarkan waktu enkripsi dan waktu dekripsi yang dimiliki masing-masing sebanyak 1024 data untuk ukuran *file* 50 KB dan 100 KB serta 170 data untuk ukuran *file* lainnya sehingga akan muncul nilai Mean Rank dan nilai signifikansi.

Langkah awal dalam pengujian Kruskal-Wallis yaitu menguji varibilitas nilai ujian data variansi ukuran *file*. Varibilitas yang dimaksud adalah bentuk dan sebaran data. Dalam hal ini, hasil yang didapatkan adalah data variansi ukuran *file* tidak memiliki sebaran data yang sama sehingga hanya dapat mengetahui perbedaan nilai *Mean*. Peringkat rata-rata waktu enkripsi SPECK128/256 ditunjukkan pada Tabel 7.148 dan waktu dekripsi ditunjukkan pada Tabel 7.149.

Tabel 7.148 Rank Kruskal-Wallis waktu enkripsi SPECK128/256

Ranks

	ukuran_file	N	Mean Rank
enkripsi	50 KB	1024	512,50
	100 KB	1024	1536,50
	150 KB	170	2133,50
	200 KB	170	2303,50
	250 KB	170	2473,50
	300 KB	170	2643,50
	Total	2728	

Tabel 7.149 Rank Kruskal-Wallis waktu dekripsi SPECK128/256

Ranks

	ukuran_file	N	Mean Rank
dekripsi	50 KB	1024	512,50
	100 KB	1024	1536,50
	150 KB	170	2133,50
	200 KB	170	2303,50
	250 KB	170	2473,50
	300 KB	170	2643,50
	Total	2728	

Langkah selanjutnya adalah analisis Kruskal-Wallis dengan pengambilan keputusan berdasarkan nilai signifikansi yang diperoleh untuk mengukur perbedaan peringkat rata-rata yang didapatkan signifikan atau tidak. Hasil pengujian Kruskal Wallis waktu enkripsi untuk SPECK128/256 akan ditunjukkan pada Tabel 7.150 dan waktu dekripsi ditunjukkan pada Tabel 7.151.

Tabel 7.150 Hasil pengujian Kruskal-Wallis waktu enkripsi SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	2437,032
df	5
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Tabel 7.151 Hasil pengujian Kruskal-Wallis waktu dekripsi SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	2437,082
df	5
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Pengambilan keputusan:

H0: nilai signifikansi > 0,05 maka tidak terdapat perbedaan waktu antara data kelompok

H1: nilai signifikansi < 0,05 maka terdapat perbedaan waktu antara data kelompok

Hasil yang didapatkan adalah nilai signifikansi < 0,05 maka keputusan H0 ditolak yang berarti terdapat perbedaan waktu antara data variansi ukuran *file* yang dianalisis. Langkah berikutnya adalah melakukan pengujian *Post Hoc test* untuk mengetahui perbandingan signifikan yang dihasilkan setiap data kelompok.

7.2.7.9 Pengujian *Post Hoc test* SPECK128/256

Pengujian *Post Hoc test* dilakukan pada 15 data kelompok dengan membandingkan masing-masing dua ukuran *file*. Pada Tabel 7.150 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk mencari

perbedaan siginifikan diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu enkripsi SPECK128/256 sebagai berikut:

1. Nilai *Chi-Square* antara 50 KB dengan 100 KB bernilai 1536,918. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.152.

Tabel 7.152 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 100 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	1536,918
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

2. Nilai *Chi-Square* antara 50 KB dengan 150 KB bernilai 438,480. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.153.

Tabel 7.153 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 150 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	438,480
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

3. Nilai *Chi-Square* antara 50 KB dengan 200 KB bernilai 438,475. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.154.

Tabel 7.154 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 200 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	438,475
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

4. Nilai *Chi-Square* antara 50 KB dengan 250 KB bernilai 438,475. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.155.

Tabel 7.155 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 250 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	438,475
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

5. Nilai *Chi-Square* antara 50 KB dengan 300 KB bernilai 438,477. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.156.

Tabel 7.156 Nilai *Chi-Square* waktu enkripsi 50 KB dengan 300 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	438,477
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

6. Nilai *Chi-Square* antara 100 KB dengan 150 KB bernilai 437,979. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.157.

Tabel 7.157 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 150 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	437,979
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

7. Nilai *Chi-Square* antara 100 KB dengan 200 KB bernilai 437,975. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.158.

Tabel 7.158 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 200 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	437,975
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

8. Nilai *Chi-Square* antara 100 KB dengan 250 KB bernilai 437,974. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.159.

Tabel 7.159 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 250 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	437,974
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

9. Nilai *Chi-Square* antara 100 KB dengan 300 KB bernilai 437,976. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.160.

Tabel 7.160 Nilai *Chi-Square* waktu enkripsi 100 KB dengan 300 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	437,976
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

10. Nilai *Chi-Square* antara 150 KB dengan 200 KB bernilai 254,575. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.161.

Tabel 7.161 Nilai *Chi-Square* waktu enkripsi 150 KB dengan 200 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	254,575
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

11. Nilai *Chi-Square* antara 150 KB dengan 250 KB bernilai 254,555. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.162.

Tabel 7.162 Nilai *Chi-Square* waktu enkripsi 150 KB dengan 250 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	254,555
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

12. Nilai *Chi-Square* antara 150 KB dengan 300 KB bernilai 254,603. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.163.

Tabel 7.163 Nilai *Chi-Square* waktu enkripsi 150 KB dengan 300 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	254,603
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

13. Nilai *Chi-Square* antara 200 KB dengan 250 KB bernilai 254,444. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.164.

Tabel 7.164 Nilai *Chi-Square* waktu enkripsi 200 KB dengan 250 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	254,444
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

14. Nilai *Chi-Square* antara 200 KB dengan 300 KB bernilai 254,493. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.165.

Tabel 7.165 Nilai *Chi-Square* waktu enkripsi 200 KB dengan 300 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	254,493
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

15. Nilai *Chi-Square* antara 250 KB dengan 300 KB bernilai 254,473. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.166.

Tabel 7.166 Nilai *Chi-Square* waktu enkripsi 250 KB dengan 300 KB SPECK128/256

Test Statistics[a,b]

	enkripsi
Chi-Square	254,473
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Berdasarkan hasil pengujian *Post hoc test*, didapatkan hasil bahwa setiap data kelompok yang diujikan menghasilkan nilai signifikansi < 0,05 semua yang berarti bahwa di setiap data kelompok terdapat perbedaan waktu yang signifikan. Maka dapat disimpulkan bahwa perubahan data setiap 50 KB menghasilkan perubahan waktu yang signifikan. Semisal perubahan 50 KB ke 100

KB dengan selisih ukuran 50 KB dapat menghasilkan perbedaan waktu yang signifikan dan begitu pula untuk ukuran *file* lainnya.

Selanjutnya pada Tabel 7.151 menghasilkan nilai signifikansi < 0,05 yang berarti terdapat perbedaan waktu antara data kelompok maka akan dilakukan pengujian *Post Hoc test* untuk untuk mencari perbedaan siginifikan diantara data kelompok yang diujikan. Hasil pengujian *Post Hoc test* waktu dekripsi SPECK128/256 sebagai berikut:

1. Nilai *Chi-Square* antara 50 KB dengan 100 KB bernilai 1536,995. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.167.

Tabel 7.167 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 100 KB SPECK128/128

Test Statistics[a,b]

	dekripsi
Chi-Square	1536,995
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

2. Nilai *Chi-Square* antara 50 KB dengan 150 KB bernilai 438,045. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.168.

Tabel 7.168 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 150 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	438,045
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

3. Nilai *Chi-Square* antara 50 KB dengan 200 KB bernilai 438,047. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.169.

Tabel 7.169 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 200 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	438,047
df	1
Asymp. Sig.	,000

4. Nilai *Chi-Square* antara 50 KB dengan 250 KB bernilai 438,042. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.170.

Tabel 7.170 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 250 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	438,042
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

5. Nilai *Chi-Square* antara 50 KB dengan 300 KB bernilai 438,043. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.171.

Tabel 7.171 Nilai *Chi-Square* waktu dekripsi 50 KB dengan 300 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	438,043
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

6. Nilai *Chi-Square* antara 100 KB dengan 150 KB bernilai 438,519. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.172.

Tabel 7.172 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 150 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	438,519
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

7. Nilai *Chi-Square* antara 100 KB dengan 200 KB bernilai 438,520. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.173.

Tabel 7.173 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 200 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	438,520
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

8. Nilai *Chi-Square* antara 100 KB dengan 250 KB bernilai 438,515. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.174.

Tabel 7.174 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 250 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	438,515
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

9. Nilai *Chi-Square* antara 100 KB dengan 300 KB bernilai 438,516. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.175.

Tabel 7.175 Nilai *Chi-Square* waktu dekripsi 100 KB dengan 300 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	438,516
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

10. Nilai *Chi-Square* antara 150 KB dengan 200 KB bernilai 254,571. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.176.

Tabel 7.176 Nilai *Chi-Square* waktu dekripsi 150 KB dengan 200 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	254,571
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

11. Nilai *Chi-Square* antara 150 KB dengan 250 KB bernilai 254,440. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.177.

Tabel 7.177 Nilai *Chi-Square* waktu dekripsi 150 KB dengan 250 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	254,440
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

12. Nilai *Chi-Square* antara 150 KB dengan 300 KB bernilai 254,482. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.178.

Tabel 7.178 Nilai *Chi-Square* waktu dekripsi 150 KB dengan 300 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	254,482
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

13. Nilai *Chi-Square* antara 200 KB dengan 250 KB bernilai 254,475. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.179.

Tabel 7.179 Nilai *Chi-Square* waktu dekripsi 200 KB dengan 250 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	254,475
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

14. Nilai *Chi-Square* antara 200 KB dengan 300 KB bernilai 254,517. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.180.

Tabel 7.180 Nilai *Chi-Square* waktu dekripsi 200 KB dengan 300 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	254,517
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

15. Nilai *Chi-Square* antara 250 KB dengan 300 KB bernilai 254,386. Nilai *Chi-Square* data kelompok ditunjukkan pada Tabel 7.181.

Tabel 7.181 Nilai *Chi-Square* waktu dekripsi 250 KB dengan 300 KB SPECK128/256

Test Statistics[a,b]

	dekripsi
Chi-Square	254,386
df	1
Asymp. Sig.	,000

a. Kruskal Wallis Test

b. Grouping Variable: ukuran_file

Berdasarkan hasil pengujian *Post hoc test*, didapatkan hasil bahwa setiap data kelompok yang diujikan menghasilkan nilai signifikansi < 0,05 semua yang berarti bahwa di setiap data kelompok terdapat perbedaan waktu yang signifikan. Maka dapat disimpulkan bahwa perubahan data setiap 50 KB menghasilkan perubahan waktu yang signifikan. Semisal perubahan 50 KB ke 100

KB dengan selisih ukuran 50 KB dapat menghasilkan perbedaan waktu yang signifikan dan begitu pula untuk ukuran *file* lainnya.

7.3 Analisis Pengujian *Avalanche Effect*

Pengujian *Avalanche Effect* dikatakan optimal apabila memiliki nilai berkisar 45-60% (Santoso et al., 2014). Perbedaan pada *ciphertext* yang dihasilkan saat adanya perubahan *plaintext* maupun *key* akan menyebabkan kriptanalis semakin sulit untuk melakukan penyerangan. Perhitungan *avalanche effect* menggunakan Persamaan 7.1.

Avalanche Effect = jumlah bit yang berubah / jumlah bit total * 100% (7.1)

Hasil pengujian *avalanche effect* ditunjukkan pada Tabel 7.182.

Tabel 7.182 Hasil Pengujian *Avalanche Effect*

Varian SPECK	*Avalanche Effect Plaintext*	*Avalanche Effect Key*	Rata-rata (%)
SPECK128/128	0.484	0.398	0.441
SPECK128/192	0.5234	0.5625	0.54295
SPECK128/256	0.4296	0.5	0.4648

Berdasarkan hasil pengujian *avalanche effect* didapatkan hasil bahwa SPECK128/192 mengalami perubahan pada *output* setengah dari jumlah bit secara keseluruhan. SPECK128/192 memiliki persentase rata-rata AE tertinggi sehingga tingkat keamanan yang dimiliki juga lebih tinggi dibanding varian SPECK lainnya.

BAB 8 PENUTUP

8.1 Kesimpulan

1. Implementasi SPECK untuk keamanan *file* teks memiliki tiga proses utama yaitu enkripsi, dekripsi dan *key schedule*. Proses awal adalah proses membangkitkan kunci dengan fungsi *key schedule* pada setiap *round* yang telah ditentukan berdasarkan masing-masing varian SPECK. Proses enkripsi dan dekripsi membutuhkan kunci di tiap *round* untuk menghasilkan *ciphertext* dan *plaintext*. Proses enkripsi dan dekripsi dipanggil oleh sistem melalui fitur proses enkripsi dan dekripsi kemudian sistem melakukan *load* data PDF agar file hasil enkripsi dan dekripsi dapat disimpan dalam bentuk PDF melalui fitur simpan hasil proses.

2. Hasil analisis yang dilakukan adalah pengujian waktu eksekusi dengan total rata-rata waktu enkripsi SPECK128/128 untuk *file* 50 KB sebesar1871,78 ms, SPECK128/192 untuk *file* 50 KB sebesar 1861,34 ms, SPECK128/256 untuk *file* 50 KB sebesar 1855,42 ms, SPECK128/128 untuk *file* 100 KB sebesar 4726,77 ms, SPECK128/192 untuk *file* 100 KB sebesar 4705,41 ms, SPECK128/256 untuk *file* 100 KB sebesar 4695,99 ms, SPECK128/128 untuk *file* 150 KB sebesar 7911,72 ms, SPECK128/192 untuk *file* 150 KB sebesar 7860,21 ms, SPECK128/256 untuk *file* 150 KB sebesar 7756,15 ms, SPECK128/128 untuk *file* 200 KB sebesar 11014,42 ms, SPECK128/192 untuk *file* 200 KB sebesar 10975,57 ms, SPECK128/256 untuk *file* 200 KB sebesar 10847,12 ms, SPECK128/128 untuk *file* 250 KB sebesar 13352,89 ms, SPECK128/192 untuk *file* 250 KB sebesar 13346,61 ms, SPECK128/256 untuk *file* 250 KB sebesar 13313,04 ms, SPECK128/128 untuk *file* 300 KB sebesar 16874,47 ms, SPECK128/192 untuk *file* 300 KB sebesar 16845,98 ms, SPECK128/256 untuk *file* 300 KB sebesar 16829,52 ms sehingga didapatkan bahwa kinerja algoritme SPECK yang paling baik ada pada SPECK128/256 yang memakan waktu proses paling singkat untuk mengenkripsi dan dekripsi semua varian ukuran *file*. Dari varian ukuran *file*, didapatkan hasil bahwa semakin besar ukuran *file* yang digunakan maka semakin lama waktu untuk sistem melakukan proses enkripsi dan dekripsi. Terakhir dari segi waktu sistem yang melibatkan pembacaan *file* saat pengguna meng*upload file* ke dalam sistem menghasilkan pengujian ideal dilakukan sampai pada ukuran *file* 100 KB yang menghasilkan waktu paling lama 5 menit.

8.2 Saran

1. Sistem hanya melakukan enkripsi dan dekripsi pada *file* teks, untuk ke depannya dapat ditambahkan proses kriptografi pada *file* audio, video maupun gambar.

2. Algoritme SPECK ke depannya dapat diimplementasikan di berbagai *platform*.

3. Mempelajari dan mengimplementasikan algoritme kriptografi yang lebih terbaru ke depannya.

DAFTAR REFERENSI

Ariyanto, E., Pravitasari, T., I. & Setyorini, 2008. *Analisa Implementasi Algoritma Stream Cipher SOSEMANUK dan DICING dalam Proses Enkripsi Data*. Bandung: Institut Teknologi Telkom.

Asriyanik, 2017. *Studi Terhadap Advanced Encryption Standard (AES) dan Algoritma Knapsack dalam Pengamanan Data*. Sukabumi: Universitas Muhammadiyah Sukabumi.

Aulia, F. dan Pratomo, A.B., 2016. *Blox: Algoritma Block Cipher*. Bandung: Institut Teknologi Bandung.

Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B. & Wingers, L., 2013. *The SIMON and SPECK Families of Lightweight Block Ciphers*. USA: National Security Agency.

Febriansyah, 2013. *Analisis Varian Satu Jalan Kruskal-Wallis*. Palembang: Raden Fatah Pelembang.

Firmanesa, I.E. dan Wildan, 2016. *Uji SAC Terhadap Algoritma SPECK*. UNY: Lembaga Sandi Negara.

Ghasemi dan Zahediasl, 2012. *Normality Tests for Statistical Analysis: A Guide for Non-Statisticians*. Iran: Endocrine Research Center, Research Institute for Endocrine Sciences, Shahid Beheshti University of Medical Sciences.

Ghozali, Imam, 2012. *Aplikasi Analisis Multivariate dengan Program IBM SPSS*. Yogyakarta: Universitas Diponegoro.

Herawati, Lucky. 2016. *Uji Normalitas Data Kesehatan Menggunakan SPSS*. Yogyakarta: Politeknik Kesehatan Kemenkes Yogyakarta.

Junaidi, 2015. *Statistik Uji Kruskal-Wallis*. Jambi: Universitas Jambi.

Nurjanah, E., 2016. *Aplikasi Enkripsi dan Dekripsi Menggunakan Algoritma RSA Berbasis Web*. Bandung: UIN Sunan Gunung Djati Bandung.

Pabokory, Astuti dan Kridalaksana, 2015. *Implementasi Kriptografi Pengaman Data pada Pesan Teks, Isi File Dokumen dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard*. Kalimantan Timur: Universitas Mulawarman.

Pratama, E.C., Nasution, S.M. dan Purboyo, T.W., 2015. *Perancangan dan Implementasi Secure Cloud dengan Menggunakan Diffie-Hellman Key Exchange Dan Serpent Cryptography Algorithm*. Bandung: Universitas Telkom.

Primartha, R., 2013. *Penerapan Enkripsi dan Dekripsi File Menggunakan Algoritma Advanced Encryption Standard (AES)*. Palembang: Universitas Sriwijaya.

Salemba, P., 2014. *Statistik untuk Kedokteran dan Kesehatan*. Jakarta: Salemba Medika.

Santoso, L., W., Budhi, G., S. & Sutanto, L., 2014. *Perbandingan Aplikasi Menggunakan Metode CAMELLIA 128 Bit Key dan 256 Bit Key*. Surabaya: Universitas Kristen Petra.

Santoso, Singgih, 2014. *Panduan Lengkap SPSS Versi 20 Edisi Revisi*. Jakarta: Elex Media Komputindo.

Shoim, A., Irfan, A. A. dan Pranoto, 2017. *Aplikasi Kriptografi Enkripsi Dekripsi File Teks Menggunakan Metode MCRYPT Blowfish*. Tuban: Universitas PGRI Ronggolawe Tuban.

Togan, M., Lupascu, C. dan Plesca, C., 2015. *Homomorphic Evaluation of SPECK Cipher*. Romania: The Publishing House of The Romanian Academy.

Vadaviya, D., O. & Tandel, P., H., 2015. *Study of Avalanche Effect In AES*. India: Uka Tarsadiya University.